Searching for conical intersections in MNDO

From Thielopedia

Jump to: navigation, search

The purpose of this tutorial is to demonstrate how to search for a minimum energy crossing point (MECP) in MNDO using semiempirical configuration interaction.

We will use the methaniminium cation ($CH_2NH_2^+$) as an example system. Some of the steps below are not strictly necessary for such a simple molecule, but they will be invaluable for larger systems.

IMPORTANT: This tutorial is written for development versions of MNDO dated later than 27 November 2007. Do not try to use an earlier version!

Contents

[hide]

[edit] Determining the active space

Before we can optimise the methaniminium MECP, we need a good starting geometry and choice of active space.

The first step is to determine the active space in the ground state. To do this we run an ordinary semiempirical calculation at any reasonable (i.e. planar) geometry. Our input file is:

```
iop=-6 jop=-1 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 +
kharge=1 imult=1 iuhf=-1
Methaniminium OM2 single point calc for orbitals
from Lischka's planar S0 min geometry - fig1a
 6   0.0   0      0.0 0     0.0 0    0 0 0
 7   1.288 1      0.0 0     0.0 0    1 0 0
 1   1.090 1    119.3 1     0.0 0    1 2 0
 1   1.019 1    121.5 1     0.0 1    2 1 3
 1   1.090 1    119.3 1   180.0 1    1 2 4
 1   1.019 1    121.5 1   180.0 1    2 1 3
 0   0.0   0      0.0 0     0.0 0    0 0 0
```

Points to note about this job:

- We are using the OM2 Hamiltonian (`iop=-6`). This is somewhat slower than AM1 but more accurate.
- The default OM2 SCF precision is not very tight and can lead to noisy gradients. This is particularly undesirable for conical intersection optimisations. The options `icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001` ensure that the gradient is accurate.
- ROHF orbitals are calculated (`imult=1 iuhf=-1`), even though methaniminium is a closed-shell system. This is to avoid convergence problems later on (see below).

The option `nsav13=2` outputs a Molden-readable file which we can use to examine the orbitals. The standard choice of active space for methaniminium is two occupied orbitals of B_2 symmetry, one occupied orbital of B_1 symmetry (π) and one unoccupied orbital of B_1 symmetry (π^*). In Molden you can see that this corresponds to orbitals 3, 5, 6 and 7.

[edit] Ground state optimisation

The next step is to optimise the ground state structure using semiempirical configuration interaction (OM2/GUGA-CI).

```
iop=-6 jop=0 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 +
iprec=100 +
kharge=1 imult=1 +
kci=5 ioutci=2 imomap=2 maxrtl=200 +
movo=1 ici1=4 ici2=0 nciref=3 mciref=0 levexc=2 iroot=4
Methaniminium OM2/GUGA-CI ground state opt
from Lischka's planar S0 min geometry - fig1a
 6   0.0   0      0.0 0     0.0 0    0 0 0
 7   1.288 1      0.0 0     0.0 0    1 0 0
 1   1.090 1    119.3 1     0.0 0    1 2 0
```

```
 1   1.019 1    121.5 1     0.0 0    2 1 3
 1   1.090 1    119.3 1   180.0 0    1 2 4
 1   1.019 1    121.5 1   180.0 0    2 1 3
 0   0.0   0      0.0 0     0.0 0    0 0 0
  3  5  6  7
```

Points to note about this job:

- There is no need to optimise the dihedral angles. In fact, allowing these to be optimised for a planar system can hinder convergence.
- For validation purposes the convergence threshold for optimisation has been set to be very tight (`iprec=100`). For everyday calculations this is not necessary.
- GUGA-CI is specified with `kci=5`.
- It is not necessary to specify `iuhf=-1` when using GUGA-CI; open-shell calculations are ROHF by default.
- The active space is explicitly defined (`movo=1`). `ici1=4 ici2=0` specifies that there are four occupied orbitals (two of which are singly-occupied) and no unoccupied orbitals. This is different from what we would naturally expect (3/1) because we are using ROHF orbitals. The actual orbitals are listed on a line after the geometry.
- `nciref=3 mciref=0` specifies that there are 3 reference configurations for the CI expansion, corresponding to the ground state and single and double HOMO-LUMO excitations. This ensures that we are doing a ground state optimisation even though we are using ROHF orbitals.
- `levexc=2` specifies that single and double CI excitations from the reference configurations are considered. It is not usually worth going to higher level excitations.
- MO orbital tracking has been switched on (`imomap=2`). This means that the active space should remain consistent over the course of the optimisation, even if the ordering of the MOs changes.

As we already have a reasonable starting geometry the optimisation should converge very quickly to a CI heat of formation of 170.84428 kcal/mol.

[edit] Distortion towards the conical intersection

Often, especially for larger systems, there will be more than one MECP accessible from the ground state geometry. It is therefore not a good idea to start an MECP optimisation from the ground state minimum (unless you are merely trying to explore the PES). Instead, you should apply an appropriate distortion towards the conical intersection you are looking for.

In our case, we are looking for a conical intersection where the C=N bond is twisted. Therefore an appropriate distortion is a twist of approximately 90 degrees around the bond (but not exactly 90 degrees, as this would restrict the symmetry of the molecule more than we want).

The MOs at this geometry will obviously be different, so we need to do another single-point OM2 calculation to determine the new active space.

```
iop=-6 jop=-1 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 +
kharge=1 imult=1 iuhf=-1
Methaniminium OM2 ROHF single point energy
at OM2/GUGA-CI ground state opt geom with c.90 deg torsion
```

```
   6     0.00000  0       0.00000  0       0.00000  0        0    0    0
   7     1.28815  1       0.00000  0       0.00000  0        1    0    0
   1     1.09516  1     121.76616  1       0.00000  0        1    2    0
   1     1.02074  1     122.65946  1     -89.0      1        2    1    3
   1     1.09516  1     121.76616  1      91.0      1        1    2    4
   1     1.02074  1     122.65946  1      91.0      1        2    1    3
   0     0.0      0       0.0      0       0.0      0        0    0    0
```

Points to note about this job:

- It is easy to apply a distortion if you're working in internal coordinates.
- The active space is easy to identify by inspection as 3, 5, 6, 7 (i.e. unchanged). With a smaller distortion this would be harder (and would also cause problems for the MO tracking algorithm later on).

[edit] Conical intersection optimisation

We now have a good starting geometry with a sensible active space so we're ready to optimise the MECP.

There are three algorithms implemented in MNDO to optimise conical intersections. In almost all cases the best algorithm to use is Lagrange-Newton.

```
iop=-6 jop=0 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 +
icross=5 ncigrd=2 ief=1 lrscal=1 dmax=0.1 iprec=100 +
kharge=1 imult=1 +
kci=5 ioutci=2 imomap=2 maxrtl=200 +
movo=1 ici1=4 ici2=0 nciref=3 mciref=0 levexc=2 iroot=4
Methaniminium Lagrange-Newton CI search
from OM2/GUGA-CI ground state opt geom with c.90 deg torsion
   6     0.00000  0       0.00000  0       0.00000  0        0    0    0
   7     1.28815  1       0.00000  0       0.00000  0        1    0    0
   1     1.09516  1     121.76616  1       0.00000  0        1    2    0
   1     1.02074  1     122.65946  1     -89.0      1        2    1    3
   1     1.09516  1     121.76616  1      91.0      1        1    2    4
   1     1.02074  1     122.65946  1      91.0      1        2    1    3
   0     0.0      0       0.0      0       0.0      0        0    0    0
  3  5  6  7
   1   2
   0.00010   1.00000
   0
```

Methaniminium geometry at the MECP. The CH_2 group is slightly pyramidalised.

Points to note about this job:

- The CI options are the same as for the ground state optimisation.

- `icross=5` selects the Lagrange-Newton algorithm. The optimiser `ief=1` must always be used with Lagrange-Newton.
- `lrscal=1 dmax=0.1` selects a 0.1 Å trust radius. A trust region of roughly this size is needed to give an efficient balance between energy minimisation and finding the CI seam.
- Again, `iprec=100` is set for validation purposes to ensure that the conical intersection is converged to under 0.00001 kcal/mol. For everyday calculations this level of convergence is not necessary and may not even be possible due to noisy gradients.
- `ncigrd=2` is needed to indicate that two CI gradients are calculated. The states involved (here, 1 and 2, corresponding to S_0 and S_1) are specified on the line following the active space orbitals.
- The extra line `0.00010 1.00000` determines the orthogonalisation thresholds t_1 and t_2 (see the paper for more details). In general these values should not be changed from the defaults.
- The final `0` indicates that no additional geometrical constraints are wanted.

This job should optimise very quickly (about 24 cycles) to a conical intersection at 246.72870 kcal/mol.

[edit] Optimisation with other algorithms

If you want to do an optimisation with one of the other algorithms, a few keywords have to be changed. The following input file uses the penalty function method:

```
iop=-6 jop=0 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 +
icross=3 ncigrd=2 iconv=3 nrst=-1 ldrop=100 iprec=100 +
kharge=1 imult=1 +
kci=5 ioutci=2 imomap=2 maxrtl=200 +
movo=1 ici1=4 ici2=0 nciref=3 mciref=0 levexc=2 iroot=4
Methaniminium Lagrange-Newton CI search
from OM2/GUGA-CI ground state opt geom with c.90 deg torsion
   6     0.00000  0       0.00000  0       0.00000  0        0    0    0
   7     1.28815  1       0.00000  0       0.00000  0        1    0    0
   1     1.09516  1     121.76616  1       0.00000  0        1    2    0
   1     1.02074  1     122.65946  1     -89.0      1        2    1    3
   1     1.09516  1     121.76616  1      91.0      1        1    2    4
   1     1.02074  1     122.65946  1      91.0      1        2    1    3
   0     0.0      0       0.0      0       0.0      0        0    0    0
  3  5  6  7
   1   2
   5.00000   5.00000
```

Changes from Lagrange-Newton:

- `icross=3` selects the penalty function method.
- The default optimiser is used, not `ief=1`. This means that `lrscal` and `dmax` are also not relevant. The penalty function method uses a line search, not a trust region.
- `nrst=-1 ldrop=100` are technical options relating to Hessian resets which increase the efficiency of the penalty function method. See the MNDO manual for more details.
- The final line `5.00000 5.00000` sets the adjustable parameters for the penalty function method (these are actually the default values). They are a compromise between robustness of

the optimisation and tightness of convergence and should usually not be changed.

The penalty function run should converge in about 110 cycles. The final state 1 heat of formation should be 246.62462 kcal/mol and state 2 should be 246.70464 kcal/mol. The gap between the states is normal and does not indicate that convergence has failed.

The next input file uses the gradient projection method:

```
iop=-6 jop=0 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 +
icross=4 ncigrd=2 iprec=10000 +
kharge=1 imult=1 +
kci=5 ioutci=2 imomap=2 maxrtl=200 +
movo=1 ici1=4 ici2=0 nciref=3 mciref=0 levexc=2 iroot=4
Methaniminium Lagrange-Newton CI search
from OM2/GUGA-CI ground state opt geom with c.90 deg torsion
   6     0.00000  0       0.00000  0       0.00000  0        0    0    0
   7     1.28815  1       0.00000  0       0.00000  0        1    0    0
   1     1.09516  1     121.76616  1       0.00000  0        1    2    0
   1     1.02074  1     122.65946  1     -89.0      1        2    1    3
   1     1.09516  1     121.76616  1      91.0      1        1    2    4
   1     1.02074  1     122.65946  1      91.0      1        2    1    3
   0     0.0      0       0.0      0       0.0      0        0    0    0
  3  5  6  7
   1   2
   1.00000   0.90000   0.10000   0.00000
```

Changes from Lagrange-Newton:

- `icross=4` selects the gradient projection method.
- Again, the default optimiser is used, not `ief=1`.
- Because each method has different criteria for convergence, the nominal convergence threshold `iprec` is not equivalent between methods. To obtain the same (very tight) level of convergence as before, `iprec` has to be raised by a factor of roughly 10-100.
- The first two numbers of the final line `1.00000 0.90000` describe the adjustable parameters of the gradient projection method (set at the defaults). They shouldn't usually be changed. The third number `0.10000` is a trust radius in Å. The fourth number scales the step size but should usually be switched off (0.0).

The gradient projection run should converge in about 36 cycles to the same value as the Lagrange-Newton run (within +/-0.00001 kcal/mol).

[edit] Singlet/triplet optimisations

Optimisations are also possible between two states with different multiplicities.

- `multci=-1` should be set to indicate that the states have different multiplicities.
- The multiplicities of the CI states should be given on an extra line following the list of states.
- Both states are calculated using a common set of ROHF orbitals (`imult=1`).

Example: Benzene S_0/T_1 optimisation. This run should optimise to 97.56889 kcal/mol in about 36 cycles.

[edit] Pitfalls

Most problems occur because something has gone wrong with the active space.

- A failure of MO tracking is the most common problem. MOs are tracked by an orbital overlap criterion, and this can sometimes fail, e.g. when a large geometry step is taken or when orbitals mix. The optimiser will try to recover by taking a smaller step, but after a few failed attempts to do this the optimiser will give up.
- Similarly, the MO tracking may appear to succeed but the active space has actually qualitatively changed. This can happen after orbitals have mixed. **You must check the active space at the end of a conical intersection optimisation to check that it hasn't changed!**

In both these cases you may be able to avoid the problem with a different starting geometry. If it still fails in the same way, you should try optimising initially with a smaller active space, then reoptimising after convergence with the original active space.

For conjugated pi systems, an alternative scheme (`movo=-5`) is available for selecting the active space which avoids the problems of MO tracking. This requires a later CVS version (3 September 2008 or later).

- The calculation may fail completely if an orbital outside the active space becomes nearly degenerate with one inside the active space.

Again, try a different starting geometry, or add the problem orbital to the active space.

- SCF convergence can fail. This is much less common with ROHF orbitals than RHF, but can still happen. You may notice that the DIIS procedure oscillates between two very close solutions.

This problem is really a limitation of DIIS. Increasing the maximum number of SCF iterations with `kitscf` may help. You could also try relaxing the SCF convergence criteria a little (try `iscf=10 iplscf=10).`

- The calculation may not fail, but it does not converge either.

Is your convergence criterion too tight? If not, try a different optimisation algorithm or try playing with the adjustable parameters. If all else fails, try a different active space or even a different Hamiltonian. It is possible that an MECP does not exist on the PESs you are calculating.

In general the best strategy is to first optimise with the smallest active space you think you can get away with, then reoptimise later with a larger one. For larger systems this is a more efficient strategy anyway.

[edit] Further Reading

- Comparison of algorithms for conical intersection optimisation using semiempirical methods

  T.W. Keal, A. Koslowski, and W. Thiel, *Theor. Chem. Acc.*, 2007, **118**, 837.

  This article discusses the three MECP optimisation algorithms implemented in MNDO and evaluates their performance. The methaniminium calculation in Table 1 was carried out as described above.

- Ultrafast two-step process in the non-adiabatic relaxation of the $CH_2NH_2^+$ molecule

  M. Barbatti, A.J.A. Aquino, and H. Lischka, *Mol. Phys.*, 2006, **104**, 1053.

  A higher level ab initio optimisation for comparison.

Non-adiabatic dynamics with MNDO

From Thielopedia

Jump to: navigation, search

The development version of MNDO (from the CVS repository) contains a dynamics driver. It is intended for small molecule studies using semiempirical configuration interaction. In particular, multiple state non-adiabatic dynamics is supported with the Tully surface hopping and Ehrenfest algorithms. Single-state (Born-Oppenheimer) dynamics is also possible.

[edit] MD scripts and tutorial

Several scripts are provided in the `mdtools/` directory of the MNDO distribution to make surface hopping calculations more convenient.

Script	Purpose
mdsample.py	Samples snapshots from a Molden or XYZ trajectory.
mdfilter.py	Selects starting geometries for trajectories from a sample XYZ file according to various criteria. Can also be used to simulate absorption spectra.
MDextr.py	Extracts MD and hopping data from MNDO output files.
mmean.py	Averages MD and hopping data.
geoman.py	Analyses trajectory files.
mndotools.py	Library for general processing of MNDO/XYZ files.

A fully worked tutorial for an example surface hopping calculation using these scripts is available in the same directory under the name `mdtools.doc`.

General questions

From Thielopedia

Jump to: navigation, search

[edit] * General questions

(1) What calculations can we do with MNDO for the excited-state problems?

```
(1) geometry optimization of the minima of ground and excited states.
(2) Search the minimum-energy points of conical-intersections.
(3) Calculate the on-the-fly surface-hopping dynamics.
```

(2) Why shall we do the electronic-structure calculations before the dynamics studies?

```
We have to perform the validation of the semiempirical methods.
```

(3) What electronic-structure theory shall we use for the excited-state calculations?

```
In this moment, the standard approach is OM2/GUGA-CI.
```

(4) Any important references for the OM2/GUGA-CI method?

```
(1) W. Weber, Ph.D. Thesis, Universitaet Zurich, Switzerland, 1996.
(2) W. Weber, W. Thiel, Theor. Chem. Acc. 103, 495 (2000).
(3) A. Koslowski, M. E. Beck, W. Thiel, J. Comput. Chem. 24, 714 (2003).
To be continue...
```

(5) What limit do we have within OM2/GUGA-CI method?

```
(1) OM2 only use STO-3G as the basis set.
(2) Be care of the long-range interaction, like hydrogen bond and
dispersion interactions.
(3) The parameters are only available for some atoms, for instance the C,
H, N, O in OM2 method within MNDO.
(4) Some systems can not be described by MNDO.
Therefore, the validation works are very important.
```

(6) What is conical intersection? What is adiabatic representation?

```
Please see book
   Conical intersections, edited by W. Domcke, D. Yarkony and H. Koeppel.
    To be continue...
```

(7) How to search conical intersections:

```
Please see
   T. W. Keal, A. Koslowski, W. Thiel, Theor. Chem. Acc. 118, 837 (2007).
    To be continue...
```

(8) What hopping methods could we use for the surface-hopping calculations?

```
There are several methods to calculate the hopping dynamics. Some of them have
been implemented inside of MNDO. See references
(1) J. C. Tully, J. Chem. Phys. 93, 1061 (1990).
(2) S. Hammes-Schiffer, J. C. Tully, J. Chem. Phys. 101, 4657 (1994).
(3) E. Fabiano, T. W. Keal, W. Thiel, Chem. Phys. 349, 334 (2008).
(4) M. Barbatti, G. Granucci, M. Persico, M. Ruckenbauer, M. Vazdar, M.
Eckert-Maksic,
```

H. Lischka, J. Photochem. Photobiol. A **190**, 228 (2007).
To be continue...

(9) Which version of MNDO shall we use for these calculations?

(1) Download the newest version from CVS account and compile it.
(2) Or you can find the MNDO version from my directory.
/ds20th/san/users/lan/program/mndo

(10) What should we do if I wish to compile my own MNDO version?

(1) Please ask other people which compiler is correct.
(2) Try to read the instructions from the Thiel wikipage.
(3) After compiling, please make SURE that you run ALL validation jobs!!!!!!!!!!!!!!!!!!!!
Sometime with different complier, the big problems appear for the testing jobs, particular for GUGA-CI and surface-hopping jobs.
Be careful!!!
(4) A few months ago, Dr. Axel Koslowski fixed a bug in MNDO.
If you have the old version, please download the newest version of MNDO from your CVS account.
Otherwise, maybe most of your testing jobs on GUGA-CI will be wrong!!!!!

(11) Where to start the calculations?

(1) Please read the instructions by T. Keal.
"Searching for conical intersections in MNDO"
Try to make sure that you understand everything.
(2) Read the present instruction in details.
(3) Ask some input file from other people in the group.

(12) To be continue...

Electronic-structure calculations

From Thielopedia

Jump to: navigation, search

Before reading this part, please make sure that you have already read the all previous instructions

"Searching for conical intersections in MNDO"

written by T. Keal. In all discussions of this page, I assume that the ground-state wafefunction is the close-shell wavefunction.

(1) Before starting, please spend some time to understand what is adiabatic and diabatic representations, see book

"Conical Intersections" edited by W. Domcke, D. Yarkony and H. Koeppel.

General speaking, adiabatic electronic wavefunctions are the eigenstates of the electronic Hamiltonian. They are not smooth functions of nuclear geometry. In adiabatic representation, the electronic state S0, S1 and S2 are named according to the order of their energies. Sometime, these wavefunctions show mixing of different electronic characters, for instance (npi*/pipi*). In adiabatic representation, the interstate coupling (nonadiabatic couplings or kinetic couplings) goes to infinity at conical intersections. On the other hand, if we follow the electronic character and define the electronic wavefunction, we have the smooth electronic wavefunctions of the nuclear geometry. They are not eigenstate of the electronic Hamiltonian. But the interstate coupling become the potential couplings which are smooth functions of nuclear geometry. This is so-called diabatic representations. In this instroduction, I always use adiabatic representations and use S0, S1 and S2 to label the electronic states

(2) It is very important to take the reasonable structures for all optimization works.

Without a good starting geometry, it is not possible to find a conical intersection. In general, please do NOT begin S0-S1 conical intersection search from the equilibrium geometry, except that there is a direct barrierless decay path connecting the S0S1 conical intersection and the equilibrium geometry.

(3) Please check text book to understand how to select the active spaces.

(4) At the OM2 level, I recommend to use ROHF (restricted-open-shell HF) for the excited-state calculations.

In principle, the close-shell HF (RHF) gives better description of the ground-state wavefunctions. After the excitations, the molecular orbitals should relax to describe the the excited states. Therefore, the ground and excited states in principle should have different orbitals. If the ground-state is the close-shell system, the excited-state should become the open-shell system. Therefore, it is better to use ROHF to describe the excited states. Of course, this means that we can not describe the ground-state very well.

The keyword to switch between ROHF or RHF is *imult=0* or *imult=1*

In standard CASSCF calculations, the natural orbitals are optimized in the state-averaged calculations. Therefore, the final orbitals can not give the best descriptions for both of the ground-state and excited-state wavefunctions, while they provide the balanced description for both of them.

(5) We have many ways to specify the active space in GUGA-CI. For all options, please find examples under this instructions. For details, please read manual.

(5.1) The keyword to active GUGA-CI is *kci=5*.
(5.2) The active space is defined by the keywords *ici1* and *ici2*.
For example the keywords *ici1=5, ici2=4* mean that the active space consists
of 5 occupied orbitals and 4 unoccupied orbitals.
(5.3) By using keyword "movo=0", the active space can be selected automatically according to their energies.
(5.4) By using keyword "movo=1", the active space can be selected manually.

Please add a line to specify the active space at the end of the input files.
(5.5) By using keyword "movo=4", the active space can be selected automatically according
to their contributions on the important references for the configuration expansions.
Please see manual for details. However, please be caution to use this method. Sometime, it gives wrong results.
(5.6) By using keyword "movo=4", "ici1", "ici2", "kci1" and "kci2", the active space can be
selected automatically.
For example the keywords *ici1=5, ici2=4, kci1=4, kci2=3* mean that the active space consists
of 5 occupied orbitals and 4 unoccupied orbitals. Among them. 4 pi and 3 pi* orbitals are selected.
(5.7) By using keyword "movo=-5", "pipop>0", the fix-sized active space can be selected automatically
for the high-conjugated systems with many conjugated CC double bonds.
In this situation, please specify the list of all conjugated atoms at the end of the input files.
(5.8) By using keyword "movo=-5", "pipop<0", the varied-sized active space can be selected automatically
for the high-conjugated systems with many conjugated CC double bonds.
In this situation, please specify the list of all conjugated atoms at the end of the input files.

(6) When you use ROHF and RHF to specify the active space, please note that the numbers of the occupied and unoccupied orbitals are different.

For example, "imult=0, ici1=5, ici2=4" is equivalent to "imult=1, ici1=6, ici2=3"

(7) In the GUGA-CI calculations, please specify the references.

(7.1) The reference can be defined automatically.
With "nciref=1 (or 2,3) mciref=0", one (or two or three) references can be constructed automatically.
(7.2) The reference can be defined from the input.
With "nciref=3 (or any number), mciref=1", the MNDO will read the references from the input file.
In this situation, please define all references at the end of the input files.
(7.3) The keyword "levexc" defines the excitation level. For instance "levexc=2" means CI-SD.

(8) In the GUGA-CI calculations, we can map the molecular orbitals during the optimization or trajectory calculations.

In general, the order of the molecular orbitals could be different at different geometries. Since all CI methods

(CASSCF, MRCI and so on) are not size-consistent method, the changing of the active space the optimization or trajectory calculations sometime results the abrupt changing of the energy. In order to keep the similar molecular orbitals inside of the active space, the method to map the molecular orbitals

are defined. The overlap of the orbitals between different geometries are calculated. The maximum overlap can define the active space in at new geometry from the active space at the old geometry.

```
(8.1) The keyword is  "imomap" and "mapthr".
(8.2) "imomap=0" means that mapping is not taken into account.
(8.3) "imomap=1" means that mapping threshold is 50%
(8.4) "imomap=2" means that mapping threshold is 90%
(8.5) "imomap=2, mapthr=70" means that mapping threshold is 70%
```

Optimization of the ground-state minimum

From Thielopedia

Jump to: navigation, search

Before start to read this section, please make sure that you have already read the all previous instructions

"Searching for conical intersections in MNDO"

written by T. Keal.

Example 1 :

```
iop=-6 jop=0 igeom=1 iform=1 +
nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 +
iplscf=11 dstep=0.00001 +
kitscf=200 +
iprint=1 mprint=1 iprec=100 +
imult=1 ioutci=1 +
kci=5  imomap=1 maxrtl=200 +
movo=1 ici1=6 ici2=3 nciref=3 +
mciref=0 levexc=2 iroot=8 iuvcd=2

OM2
  6    -0.0038158628 1     0.0245291050 1     0.0002045116 1
  6     1.4444455762 1    -0.0028495847 1     0.0025579728 1
  6     2.1003818970 1    -1.2785586395 1     0.0015963221 1
  7     1.5244423372 1    -2.5151010722 1    -0.0013462915 1
  6     0.1907596134 1    -2.4583656889 1    -0.0034641595 1
  7    -0.5573002077 1    -1.2909838270 1    -0.0028239832 1
  7     2.3827054852 1     0.9939920225 1     0.0057199715 1
  6     3.5661060171 1     0.3955068888 1     0.0067161496 1
  7     3.4418138951 1    -1.0020073821 1     0.0042534315 1
  7    -0.5067042104 1    -3.6378509876 1    -0.0065229881 1
  8    -0.7914367066 1     0.9772658979 1     0.0004289937 1
  1     4.5315690746 1     0.8994845534 1     0.0091391637 1
```

```
  1     4.1858120431 1    -1.6820551368 1     0.0043893679 1
  1    -1.5706248963 1    -1.3493085940 1    -0.0045904916 1
  1    -1.5000816208 1    -3.6759518635 1    -0.0082346686 1
  1    -0.0118748942 1    -4.5009375650 1    -0.0071392252 1
  0     0.0000000000 0     0.0000000000 0     0.0000000000 0
 24 25  26 27 28 29 30 31 32
```

Short description: The present calculation use ROHF OM2 method. Define the active space from the input.

```
"igeom=1" :   Cartesian coordinates
              If Z-matrix is given in input, please use "igeom=0". See Tomas
Keal's instructions for details.
"kitscf=200": The number of iterations in SCF calculations.
"imult=1"   : The ROHF calculation for singlet state.
"imomap=1"  : The threshold of the molecular-orbital mapping is 0.5
              This can also be defined by "imomap=2, mapthr=50".
"maxrtl=200": The maximum number of the optimizations.
"kci=5:     : GUGA-CI
"movo=1"    : Read the active space from the input files.
"ici1=6"    : The number of the occupied orbitals in the active space.
              Please note here we have 5 occupied orbitals in the close-shell
system.
              Because we use ROHF here, we define this number as 6=5+1.
"ici2=3"    : The number of the unoccupied orbitals in the active space.
               Please note here we have 4 unoccupied orbitals in the close-shell
system.
              Because we use ROHF here, we define this number as 6=5+1.
"nciref=3"  : The number of the references.
"mciref=0"  : Define the references automatically.
              Here "nciref=3", we have close-shell, HOMO->LUMO single and double
excitations.
"levexc=2"  : CI-SD.
"iroot=8"   : Eight electronic states.
"iuvcd=2"   : Print the useful information such as dipole moments and oscillator
strengthes.
The last line specifies the molecular orbitals inside of active space.
```

Example 2 :

```
iop=-6 jop=0 igeom=1 iform=1 +
nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 +
iplscf=11 dstep=0.00001 +
kitscf=200 +
iprint=1 mprint=1 iprec=100 +
imult=0 ioutci=1 +
kci=5  imomap=2 mapthr=70 maxrtl=200 +
movo=0 ici1=5 ici2=4 nciref=3 +
mciref=0 levexc=2 iroot=8 iuvcd=2
```

```
OM2
  6    -0.0038158628 1     0.0245291050 1     0.0002045116 1
  6     1.4444455762 1    -0.0028495847 1     0.0025579728 1
  6     2.1003818970 1    -1.2785586395 1     0.0015963221 1
  7     1.5244423372 1    -2.5151010722 1    -0.0013462915 1
  6     0.1907596134 1    -2.4583656889 1    -0.0034641595 1
  7    -0.5573002077 1    -1.2909838270 1    -0.0028239832 1
  7     2.3827054852 1     0.9939920225 1     0.0057199715 1
  6     3.5661060171 1     0.3955068888 1     0.0067161496 1
  7     3.4418138951 1    -1.0020073821 1     0.0042534315 1
  7    -0.5067042104 1    -3.6378509876 1    -0.0065229881 1
  8    -0.7914367066 1     0.9772658979 1     0.0004289937 1
  1     4.5315690746 1     0.8994845534 1     0.0091391637 1
  1     4.1858120431 1    -1.6820551368 1     0.0043893679 1
  1    -1.5706248963 1    -1.3493085940 1    -0.0045904916 1
  1    -1.5000816208 1    -3.6759518635 1    -0.0082346686 1
  1    -0.0118748942 1    -4.5009375650 1    -0.0071392252 1
  0     0.0000000000 0     0.0000000000 0     0.0000000000 0
```

Short descriptions: The present calculation use RHF OM2 method. Define the active space automatically.

```
"imult=0, ici1=5 ici2=4" : Close-shell OM2. 5 occupied and 4 unoccupied orbitals
in the active space.
"imomap=2 mapthr=70"     : The threshold of the molecular-orbital mapping is 0.7.
                           Only with "imomap=2", the threshold of the molecular-
orbital mapping is 0.9.
"movo=0"                 : Construct the active space by considering the energies
of molecular orbitals only.
```

Example 3

Short description: Find an example with the keywords "jci1 and jci2"

Example 4

```
iop=-6 jop=0 igeom=1 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 iplscf=11 dstep=0.00001 +
iprint=1 mprint=1 iprec=100 +
imult=1 ioutci=1 +
kci=5  imomap=1 maxrtl=200 +
movo=1 ici1=7 ici2=3  nciref=4 mciref=1 levexc=2 iroot=6 iuvcd=2

Adenine OM2
7   -0.00401300 1   -0.00164100 1    0.08900800 1
```

```
6    1.37674300 1    0.03391000 1    0.09817700 1
6    1.86804200 1    1.34070500 1    0.00562400 1
7    0.73757400 1    2.12429300 1   -0.07324000 1
6   -0.33979900 1    1.27651000 1   -0.01776200 1
7    3.15514100 1    1.72356200 1    0.00270500 1
6    3.96573800 1    0.66389500 1    0.11602700 1
7    3.64270600 1   -0.64529800 1    0.22256700 1
6    2.34452500 1   -0.98297000 1    0.21406700 1
7    2.00210400 1   -2.30302200 1    0.25238500 1
1    5.03202800 1    0.87899700 1    0.12763200 1
1    1.05063800 1   -2.51917600 1    0.52187900 1
1    2.72420300 1   -2.93367300 1    0.57786300 1
1    0.72316000 1    3.13416600 1   -0.15552100 1
1   -1.35707500 1    1.64558900 1   -0.05911800 1
0    0.00000000 0    0.00000000 0    0.00000000 0
 20  21  22  23  24  25  26  27  28  29
 2  2   2   2   2   2   0   0   0   0
 2  2   2   2   1   2   1   0   0   0
 2  2   2   2   2   1   1   0   0   0
 2  2   2   2   2   1   0   1   0   0
```

Short descriptions: The present calculation use ROHF OM2 method. Define the active space and references from the input.

```
"nciref=4 mciref=1"  : Four references are used in MRCI calculations. The
references are specify at the end of the input files.
"2"                  : Double occupied orbitals
"1"                  : Single occupied orbitals
"0"                  : Empty orbitals
```

Example 5

```
iop=-6 jop=0 igeom=0 iform=1 nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iconv=3 +
iprint=1 mprint=1 iprec=100 ktrial=11 +
imult=1 ioutci=1 kitscf=500 maxrtl=800 +
kci=5 movo=-5 ici1=8 ici2=6 jci1=8 jci2=6 pipop=4000 +
levexc=2 nciref=3 mciref=0 nconj=22 +
iroot=3 ioutci=2 iuvcd=2
Title: PPT OM2
Defaut setting: iscf=9 iplscf=9 dstep=0.00001
  6    0.000000    0    0.000000    0    0.000000    0   0   0   0
  6    1.354900    1    0.000000    0    0.000000    0   1   0   0
  6    1.493900    1  125.064300    1    0.000000    0   2   1   0
  6    1.427200    1  125.521800    1   64.056200    1   3   2   1
  6    1.444500    1  123.160700    1    1.319800    1   4   3   2
  6    1.383900    1  121.135900    1  180.188900    1   5   4   3
  6    1.436600    1  120.588300    1   -0.166000    1   6   5   4
  6    1.383300    1  120.182500    1   -1.298600    1   7   6   5
  6    1.453400    1  118.106700    1    2.143200    1   4   5   6
  6    1.430200    1  119.213900    1 -183.801500    1   9   4   5
  6    1.394900    1  120.338500    1   -1.027900    1  10   9   4
```

```
  6    1.410100    1  120.405600    1    1.970000    1  11  10   9
  6    1.524300    1  122.239200    1 -180.269900    1  12  11  10
  6    1.572000    1  111.189500    1 -228.378600    1  13  12  11
  6    1.532000    1  123.692400    1  185.000000    1   2   1   3
  6    1.555100    1  110.327800    1 -112.572900    1  15  14  13
  1    1.100200    1  110.386300    1   60.640600    1  16  15  14
  1    1.102100    1  109.629900    1  -58.820700    1  16  15  14
  1    1.099300    1  111.176500    1 -178.621800    1  16  15  14
  1    1.094700    1  108.321100    1  128.848000    1  15  14  13
  1    1.102600    1  108.838800    1  278.283500    1  14  13  12
  1    1.102700    1  108.630600    1  161.385600    1  14  13  12
  1    1.104400    1  110.590900    1    9.406400    1  13  12  11
  1    1.104400    1  109.027400    1 -108.545600    1  13  12  11
  1    1.099500    1  120.159000    1  181.394100    1  11  10   9
  1    1.098900    1  119.007700    1  180.023800    1  10   9   4
  1    1.098600    1  118.388700    1  180.743100    1   8   9   4
  1    1.098400    1  120.427800    1  179.756400    1   7   8   9
  1    1.098500    1  120.171500    1  180.326000    1   6   5   4
  1    1.096400    1  118.173700    1  183.342100    1   5   4   9
  6    1.493900    1  125.064300    1 -180.000001    1   1   2   3
  6    1.427200    1  125.521800    1   64.056200    1  31   1   2
  6    1.444500    1  123.160700    1    1.319800    1  32  31   1
  6    1.383900    1  121.135900    1  180.188900    1  33  32  31
  6    1.436600    1  120.588300    1   -0.166000    1  34  33  32
  6    1.383300    1  120.182500    1   -1.298600    1  35  34  33
  6    1.453400    1  118.106700    1    2.143200    1  32  33  34
  6    1.430200    1  119.213900    1 -183.801500    1  37  32  33
  6    1.394900    1  120.338500    1   -1.027900    1  38  37  32
  6    1.410100    1  120.405600    1    1.970000    1  39  38  37
  6    1.524300    1  122.239200    1 -180.269900    1  40  39  38
  6    1.572000    1  111.189500    1 -228.378600    1  41  40  39
  6    1.532000    1  123.692400    1  185.000000    1   1   2  31
  6    1.555100    1  110.327800    1 -112.572900    1  43  42  41
  1    1.100200    1  110.386300    1   60.640600    1  44  43  42
  1    1.102100    1  109.629900    1  -58.820700    1  44  43  42
  1    1.099300    1  111.176500    1 -178.621800    1  44  43  42
  1    1.094700    1  108.321100    1  128.848000    1  43  42  41
  1    1.102600    1  108.838800    1  278.283500    1  42  41  40
  1    1.102700    1  108.630600    1  161.385600    1  42  41  40
  1    1.104400    1  110.590900    1    9.406400    1  41  40  39
  1    1.104400    1  109.027400    1 -108.545600    1  41  40  39
  1    1.099500    1  120.159000    1  181.394100    1  39  38  37
  1    1.098900    1  119.007700    1  180.023800    1  38  37  32
  1    1.098600    1  118.388700    1  180.743100    1  36  37  32
  1    1.098400    1  120.427800    1  179.756400    1  35  36  37
  1    1.098500    1  120.171500    1  180.326000    1  34  33  32
  1    1.096400    1  118.173700    1  183.342100    1  33  32  37
  0    0           0    0           0    0           0   0   0   0
  1 2 3 4 5 6 7 8 9 10 11 12 31 32 33 34 35 36 37 38 39 40
```

Short descriptions: The present calculation use ROHF OM2 method to do the optimization for a high-conjugated systems. Therefore, the active space only contains the pi orbitals selected automatically.

```
"igeom=0"   : Input coordinate is given by Z-matrix
"movo=-5"   : The active space only include the pi-type orbitals, which are
selected automatically.
```

```
"ici1=8 ici2=6 jci1=8 jci2=6" :  7 occupied and 7 unoccuppied orbitals. (Note we
use "imult=1")
"pipop=4000" " The orbitals with pi-population larger than 0.4 is treated as pi
orbitals.
"nconj=22"  :  Because the present system is a high conjugated systems. Specify
the number of the conjugated atoms.
The last line gives all  the conjugated atoms.
```

Example 6

Short descriptions: (Find an example with movo=4)

Optimization of the excited-state minimum

From Thielopedia

Jump to: navigation, search

Please make sure that you have already read the all previous instructions

```
 [1] "Searching for conical intersections in MNDO" written by T. Keal.
 [2] "Optimization of the ground-state minimum" written by Z. Lan
```

Example 1 :

```
iop=-6 jop=0 igeom=1 iform=1 +
nsav13=2 nsav7=4 +
icuts=-1 icutg=-1 iscf=11 +
iplscf=11 dstep=0.00001 +
kitscf=200 +
iprint=1 mprint=1 iprec=100 +
imult=1 ioutci=1 +
kci=5  imomap=1 maxrtl=200 +
movo=1 ici1=6 ici2=3 nciref=3 +
mciref=0 levexc=2 iroot=8 iuvcd=2 +
lroot=2

OM2
  6    -0.0038158628 1     0.0245291050 1     0.0002045116 1
  6     1.4444455762 1    -0.0028495847 1     0.0025579728 1
  6     2.1003818970 1    -1.2785586395 1     0.0015963221 1
  7     1.5244423372 1    -2.5151010722 1    -0.0013462915 1
  6     0.1907596134 1    -2.4583656889 1    -0.0034641595 1
  7    -0.5573002077 1    -1.2909838270 1    -0.0028239832 1
  7     2.3827054852 1     0.9939920225 1     0.0057199715 1
```

```
  6     3.5661060171 1     0.3955068888 1     0.0067161496 1
  7     3.4418138951 1    -1.0020073821 1     0.0042534315 1
  7    -0.5067042104 1    -3.6378509876 1    -0.0065229881 1
  8    -0.7914367066 1     0.9772658979 1     0.0004289937 1
  1     4.5315690746 1     0.8994845534 1     0.0091391637 1
  1     4.1858120431 1    -1.6820551368 1     0.0043893679 1
  1    -1.5706248963 1    -1.3493085940 1    -0.0045904916 1
  1    -1.5000816208 1    -3.6759518635 1    -0.0082346686 1
  1    -0.0118748942 1    -4.5009375650 1    -0.0071392252 1
  0     0.0000000000 0     0.0000000000 0     0.0000000000 0
 24 25  26 27 28 29 30 31 32
```

Short descriptions: The way to active the excited-state optimization are very easy. Use keyword "lroot". The other part of the input files are same as the those for the ground-state optimizations. To understand all keywords in the input file, please see Optimization of the ground-state minimum firstly.

```
 "lroot=1":  Calculate the gradient of the ground state S0.
             In optimization jobs, "lroot=1" define the optimization on the
ground-state surface.
             This is also the default setup.
 "lroot=2":  Calculate the gradient of the first excited state S1.
             In optimization jobs, "lroot=2" define the optimization on the first
exctied-state surface.
  Following the same ideas, you can define  "lroot=3", "lroot=4" or  ...
  Please notice that the number of "iroot" should be large than that of "lroot"
```

Searching for the conical intersections

From Thielopedia

Jump to: navigation, search

In this section, I only show the searching of the conical intersection with Lagrange-Newton method. The basic theory of this method can be found in the references

```
 [1] "Conical Intersections" edited by W. Domcke, D. Yarkony and H. Koeppel.
 [2] T. W. Keal, A. Koslowski, W. Thiel, Theor. Chem. Acc. 118, 837 (2007).
```

To understand the below examples, please firstly get familiar with the basic keywords from the instructions

```
 [1] "Searching for conical intersections in MNDO" written by T. Keal.
 [2] "Optimization of the ground-state minimum" written by Z. Lan
```

Example: 1

```
IOP=-6 JOP=0 IGEOM=1 IFORM=1 NSAV13=2 NSAV7=4 +
ICUTS=-1 ICUTG=-1 ISCF=11 IPLSCF=11 DSTEP=0.00001 +
IPRINT=1 MPRINT=1 IPREC=100 +
icross=5 ief=1  ncigrd=2 lrscal=1 dmax=0.1 +
IMULT=1  IOUTCI=1  iuvcd=2 +
KCI=5  IMOMAP=2 MAXRTL=200 +
MOVO=1 ICI1=6 ICI2=3 NCIREF=3 MCIREF=0 LEVEXC=2 IROOT=8 iuvcd=2
Search the Conical Intersectin between s0 and s1
OM2
  6     -0.09197563     1    -0.05085970    1     -0.12623721  1
  6      1.31149560     1    -0.07046036    1      0.27771947  1
  6      2.05292743     1    -1.34666733    1     -0.03589189  1
  7      1.56478858     1    -2.42164506    1     -0.63941166  1
  6      0.19535177     1    -2.49945546    1     -0.40630438  1
  7     -0.50720191     1    -1.37819997    1     -0.66116754  1
  7      2.26384645     1     0.90275849    1      0.22303419  1
  6      3.50329857     1     0.36577408    1      0.21809859  1
  7      3.37664005     1    -0.97739862    1      0.16784275  1
  7     -0.20861439     1    -3.31533637    1      0.61092173  1
  8     -0.76142603     1     0.91354711    1      0.21840494  1
  1      4.42971328     1     0.93693556    1      0.52340350  1
  1      4.09595529     1    -1.57207890    1     -0.18473492  1
  1     -1.51608979     1    -1.42064004    1     -0.52113135  1
  1     -1.17540084     1    -3.60682501    1      0.69529509  1
  1      0.46390321     1    -3.83098427    1      1.11809988  1
 0     0.0000000000 0     0.0000000000 0     0.0000000000 0
 24 25  26 27 28 29 30 31 32
 1   2
 0.00010   1.00000
 0
```

Short description: The input of the conical-intersection search can be constructed by adding a few keywords based on the standard input.

```
"icross=5"  :  The keyword to active the conical-intersection search
"ncigrd=2"  :  The gradient, interstate coupling vectors for two lowest electronic
states are calculated.
"1 2"       :  At the end of the input, please specify which conical intersections
you wish to look for.
               "1 2" means the S0-S1 conical intersection.
```

Example: 2

```
IOP=-6 JOP=0 IGEOM=0 IFORM=1 NSAV13=2 NSAV7=4 +
ICUTS=-1 ICUTG=-1 ISCF=9 IPLSCF=9 DSTEP=0.00001 +
IPRINT=1 MPRINT=1 IPREC=100 +
icross=5 ief=1  ncigrd=2 lrscal=1 dmax=0.05 +
IMULT=1  ioutci=2  iuvcd=2 +
KCI=5 imomap=0  MAXRTL=600 +
MOVO=-5 ici1=8 ici2=6 jci1=8 jci2=6  pipop=4000 iroot=3 +
levexc=2 nciref=3 mciref=0 nconj=22
Search the Conical Intersectin between s0 and s1
OM2
```

6	0.0000000000	0	0.0000000000	0	0.0000000000	0	0	0	0
6	1.3291863570	1	0.0000000000	0	0.0000000000	0	1	0	0
6	1.5506176683	1	128.7595428447	1	0.0000000000	0	2	1	0
6	1.4307811789	1	121.2833169172	1	25.5276391897	1	3	2	1
6	1.4198745067	1	123.1235008383	1	4.5208806406	1	4	3	2
6	1.3753311643	1	120.6846151488	1	180.0995245180	1	5	4	3
6	1.4118263525	1	121.4991175180	1	-1.5239550411	1	6	5	4
6	1.3717297716	1	119.5718863197	1	1.4862217576	1	7	6	5
6	1.4309544893	1	117.4206300212	1	0.8515877446	1	4	5	6
6	1.4162475119	1	119.4338272345	1	-179.9278230881	1	9	4	5
6	1.3740096789	1	120.3134599278	1	-2.3131859011	1	10	9	4
6	1.4166414604	1	120.9239811511	1	2.3925829660	1	11	10	9
6	1.5038437103	1	114.8385418461	1	-175.6528139788	1	12	11	10
6	1.5121204055	1	114.2819443769	1	-167.4996078213	1	13	12	11
6	1.5513257054	1	119.5271039229	1	184.9946829587	1	2	1	3
6	1.5196889729	1	112.2089272672	1	-60.2864891382	1	15	14	13
1	1.0992754682	1	111.6034518815	1	64.7798457032	1	16	15	14
1	1.0999230186	1	109.1014664086	1	-55.5404224960	1	16	15	14
1	1.0980174925	1	109.6179701076	1	-174.5328154516	1	16	15	14
1	1.1241229002	1	109.0300001692	1	178.9484466388	1	15	14	13
1	1.1150669209	1	109.6769922651	1	197.0785887626	1	14	13	12
1	1.1252296759	1	109.6839533155	1	79.5396449949	1	14	13	12
1	1.1162895022	1	106.2816678248	1	71.3446711828	1	13	12	11
1	1.1186709605	1	109.2155771482	1	-45.0375310903	1	13	12	11
1	1.0959794638	1	120.1228041655	1	183.0174125455	1	11	10	9
1	1.0980896516	1	119.0117539456	1	177.7966632851	1	10	9	4
1	1.0972996160	1	118.9432206383	1	179.3936407187	1	8	9	4
1	1.0946797672	1	120.8496128619	1	179.1565026305	1	7	8	9
1	1.0944890836	1	119.5671379377	1	179.1315232007	1	6	5	4
1	1.0838746170	1	119.1950427597	1	181.7493576524	1	5	4	9
6	1.4214915170	1	155.1208538143	1	-90.0001800000	0	1	2	3
6	1.4347410860	1	119.6263627821	1	-11.4304426785	1	31	1	2
6	1.4184769432	1	123.0390023601	1	-0.2361817682	1	32	31	1
6	1.3748840606	1	121.5390166786	1	180.9344294314	1	33	32	31
6	1.4117935967	1	120.0592545070	1	-0.6547526343	1	34	33	32
6	1.3761515471	1	120.0382345691	1	1.8068448423	1	35	34	33
6	1.4263703238	1	118.2157103374	1	-1.5021886659	1	32	33	34
6	1.4153398922	1	119.9828709390	1	-178.2147984154	1	37	32	33
6	1.3754309097	1	119.6775110813	1	2.8030374736	1	38	37	32
6	1.4079725293	1	121.9382401467	1	-1.7668680992	1	39	38	37
6	1.4906951025	1	119.9663428841	1	-182.7159950704	1	40	39	38
6	1.5214062900	1	110.5782244062	1	-158.5835925210	1	41	40	39
6	1.5897190168	1	82.0150376124	1	161.3389366200	1	1	2	31
6	1.5240389209	1	112.5051396990	1	-50.0371200928	1	43	42	41
1	1.0924564648	1	110.1211877147	1	72.2987425855	1	44	43	42
1	1.1002671049	1	108.3997476894	1	-47.5451199896	1	44	43	42
1	1.1006167213	1	111.6576588341	1	-166.8005242606	1	44	43	42
1	1.1407316795	1	105.0664078193	1	192.4874838336	1	43	42	41
1	1.1224685550	1	109.5185169828	1	182.7606392681	1	42	41	40
1	1.1123908587	1	109.7364802730	1	62.8924321213	1	42	41	40
1	1.1236955098	1	110.7679355092	1	78.5850701853	1	41	40	39
1	1.1216089454	1	110.1986520052	1	-39.4993891201	1	41	40	39
1	1.1038197122	1	119.9410883968	1	178.5205329769	1	39	38	37
1	1.0927746983	1	119.0802591978	1	181.9917740282	1	38	37	32
1	1.0982652757	1	118.8475005977	1	178.3584619106	1	36	37	32
1	1.0967033861	1	120.8350659666	1	179.8285413108	1	35	36	37
1	1.0959245743	1	120.9128637080	1	179.4105812265	1	34	33	32
1	1.1005593417	1	118.7117976947	1	178.1196848694	1	33	32	37

```
  0     0.0000000000 0     0.0000000000 0     0.0000000000 0    0    0    0
 1 2 3 4 5 6 7 8 9 10 11 12 31 32 33 34 35 36 37 38 39 40
 1   2
 0.00010   1.00000
 0
```

Short description: This is an example to search the S0-S1 conical intersection with constrains.

```
  Please go to the line
  " 6     1.4214915170 1   155.1208538143 1   -90.0001800000 0    1    2    3".
  Here "0" after "-90.0001800000" means that this dihedral angle is fixed during
the optimization.
```

Plot the gradient vectors

From Thielopedia

Jump to: navigation, search

In many situation, you need to plot different kinds of vectors, such as the gradients of the ground and excited states, gradient difference vector g, intersection coupling vector h and the gradient sum vector s. The definition of the g, h, and s vectors can be found in books

```
"Conical Intersections" edited by W. Domcke, D. Yarkony and H. Koeppel.
```

The idea to plot these vectors are as follows. In principle, all of these vectors could be represented by the displacement vectors acting on each atom of a molecular system. On the other hand, the normal-mode displacement vectors are also the vectors acting on every atoms. Many programs, such as MOLDEN, MOLEKEL, can be used to view the normal-mode vectors. Therefore, if we replace the normal-mode vectors by gradient vectors in the input file of MOLDEN, the important vectors can be plotted.

The present task can be divided by three steps:

(1) We should firstly obtain all of these vectors at a geometry. An example of the MNDO input for this purpose is given here.

```
IOP=-6 JOP=-2 NSAV13=2 NSAV7=4 +
ICUTS=-1 ICUTG=-1 IGEOM=1 IFORM=2 +
ISCF=9 IPLSCF=9 DSTEP=0.00001 +
IPRINT=1 MPRINT=5 IPREC=100 +
ICROSS=2  NCIGRD=3  +
IMULT=1  IOUTCI=2  IUVCD=3 +
kci=5  imomap=1 maxrtl=200 +
movo=1 ici1=7 ici2=3  nciref=4 mciref=1 +
levexc=2 iroot=3

Adenine OM2
  7    -0.0030315513 1    -0.0006125271 1     0.1027355406 1
```

```
  6     1.3256381174 1    -0.0101641412 1     0.1302861885 1
  6     1.8650728319 1     1.3666737393 1    -0.0049772635 1
  7     0.7312773885 1     2.1651570565 1    -0.1123228144 1
  6    -0.3719034978 1     1.3072494761 1    -0.0433430840 1
  7     3.1365110197 1     1.6927990574 1    -0.0142111144 1
  6     4.0111162303 1     0.6417584324 1     0.1147191670 1
  7     3.6726895744 1    -0.6608582372 1     0.2461123474 1
  6     2.3623380699 1    -1.0080293020 1     0.2567804458 1
  7     2.0311289919 1    -2.3144535852 1     0.3886624038 1
  1     5.0872366545 1     0.8875343017 1     0.1101398201 1
  1     1.0733457172 1    -2.6055584683 1     0.4004138669 1
  1     2.7403461095 1    -3.0150948098 1     0.4770542767 1
  1     0.7180900709 1     3.1659316826 1    -0.2186562244 1
  1    -1.4009324886 1     1.6481396862 1    -0.1000589488 1
  0     0.0000000000 0     0.0000000000 0     0.0000000000 0
 20  21  22  23  24  25  26  27  28  29
  2   2   2   2   2   2   0   0   0   0
  2   2   2   2   1   2   1   0   0   0
  2   2   2   2   2   1   1   0   0   0
  2   2   2   2   2   1   0   1   0   0
 1 2 3
```

Here a single point calculation is performed with the keyword "MPRINT=5", which will define the output level to include all gradients of each states (S0, S1, S2) and the interstate couplings (h01, h02, h12).

(2) We should extract the reference structures and all vectors. In the below example, I only extract the gradient vectors of the S0, S1 and S2 states based on the output of the above input for illustration.

```
----------------------------------------
 Molecular Geometry
----------------------------------------
  N     1    7            -0.0030315513       -0.0006125271        0.1027355406
  C     2    6             1.3256381174       -0.0101641412        0.1302861885
  C     3    6             1.8650728319        1.3666737393       -0.0049772635
  N     4    7             0.7312773885        2.1651570565       -0.1123228144
  C     5    6            -0.3719034978        1.3072494761       -0.0433430840
  N     6    7             3.1365110197        1.6927990574       -0.0142111144
  C     7    6             4.0111162303        0.6417584324        0.1147191670
  N     8    7             3.6726895744       -0.6608582372        0.2461123474
  C     9    6             2.3623380699       -1.0080293020        0.2567804458
  N    10    7             2.0311289919       -2.3144535852        0.3886624038
  H    11    1             5.0872366545        0.8875343017        0.1101398201
  H    12    1             1.0733457172       -2.6055584683        0.4004138669
  H    13    1             2.7403461095       -3.0150948098        0.4770542767
  H    14    1             0.7180900709        3.1659316826       -0.2186562244
  H    15    1            -1.4009324886        1.6481396862       -0.1000589488
 ------------------------------------------------
 Gradient 1
------------------------------------------------
                        COORDINATES (ANGSTROM)              GRADIENTS (KCAL/
(MOL*ANGSTROM))

       I   NI          X           Y           Z               X           Y
Z
```

```
       1    7       -0.00303    -0.00061     0.10274         55.91711   -31.12043
4.41771
       2    6        1.32564    -0.01016     0.13029        -80.46746    -7.59738
-0.79585
       3    6        1.86507     1.36667    -0.00498         50.47507    36.70894
-2.88860
       4    7        0.73128     2.16516    -0.11232         11.37915    24.87716
-2.41602
       5    6       -0.37190     1.30725    -0.04334        -14.72338    13.72362
-1.75289
       6    7        3.13651     1.69280    -0.01421        -74.20588    17.82824
-3.37119
       7    6        4.01112     0.64176     0.11472         51.62631    -8.72478
1.95392
       8    7        3.67269    -0.66086     0.24611         -0.78688   -28.00000
2.95458
       9    6        2.36234    -1.00803     0.25678         22.44722   -17.86325
2.34330
      10    7        2.03113    -2.31445     0.38866          1.24875    13.25522
-1.38108
      11    1        5.08724     0.88753     0.11014         -8.16176    -1.75788
0.02457
      12    1        1.07335    -2.60556     0.40041         -4.78721    -0.86693
-0.00347
      13    1        2.74035    -3.01509     0.47705          4.09073    -4.12941
0.51845
      14    1        0.71809     3.16593    -0.21866          3.16840   -14.08636
1.56076
      15    1       -1.40093     1.64814    -0.10006        -17.22018     7.75327
-1.16418
 ------------------------------------------------------------
 Gradient 2
 ------------------------------------------------------------
                           COORDINATES (ANGSTROM)               GRADIENTS (KCAL/
(MOL*ANGSTROM))

       I   NI          X           Y           Z               X           Y
Z

       1    7       -0.00303    -0.00061     0.10274        -13.09273   -15.43393
2.11085
       2    6        1.32564    -0.01016     0.13029         -7.35670     9.88811
-1.35729
       3    6        1.86507     1.36667    -0.00498         27.61692   -32.20545
2.78498
       4    7        0.73128     2.16516    -0.11232         -4.23681    27.12221
-2.28358
       5    6       -0.37190     1.30725    -0.04334         36.16741    -7.37814
0.71301
       6    7        3.13651     1.69280    -0.01421          0.59136    39.18909
-3.07086
       7    6        4.01112     0.64176     0.11472        -45.48160     1.39106
-2.17494
       8    7        3.67269    -0.66086     0.24611         53.01371   -32.75602
5.65739
       9    6        2.36234    -1.00803     0.25678        -31.74465     0.24989
-0.60540
```

```
      10    7        2.03113    -2.31445     0.38866         12.93232    20.58453
-2.07194
      11    1        5.08724     0.88753     0.11014         -4.28370    -0.77823
-0.07603
      12    1        1.07335    -2.60556     0.40041         -5.80533    -0.72532
0.03623
      13    1        2.74035    -3.01509     0.47705          4.36092    -5.48844
0.38534
      14    1        0.71809     3.16593    -0.21866          2.54466   -15.18374
1.60601
      15    1       -1.40093     1.64814    -0.10006        -25.22579    11.52438
-1.65378
-------------------------------------------------------------
Gradient 3
-------------------------------------------------------------
                              COORDINATES (ANGSTROM)              GRADIENTS
(KCAL/(MOL*ANGSTROM))

       I   NI          X           Y           Z               X           Y
Z

       1    7       -0.00303    -0.00061     0.10274          1.26176     1.40106
-0.85648
       2    6        1.32564    -0.01016     0.13029          0.69916    -0.92765
0.27883
       3    6        1.86507     1.36667    -0.00498         -2.65478     3.18244
0.78660
       4    7        0.73128     2.16516    -0.11232          0.41364    -2.64946
-0.39037
       5    6       -0.37190     1.30725    -0.04334         -3.46207     0.77676
0.63337
       6    7        3.13651     1.69280    -0.01421         -0.03478    -3.83823
-0.66670
       7    6        4.01112     0.64176     0.11472          4.31720    -0.02651
1.20698
       8    7        3.67269    -0.66086     0.24611         -5.03254     3.01669
-1.54331
       9    6        2.36234    -1.00803     0.25678          3.02453    -0.02856
0.00838
      10    7        2.03113    -2.31445     0.38866         -1.23505    -1.95261
0.32899
      11    1        5.08724     0.88753     0.11014          0.40896     0.08265
0.07319
      12    1        1.07335    -2.60556     0.40041          0.55237     0.06252
-0.07071
      13    1        2.74035    -3.01509     0.47705         -0.42006     0.55119
0.21599
      14    1        0.71809     3.16593    -0.21866         -0.24257     1.45422
-0.10087
      15    1       -1.40093     1.64814    -0.10006          2.40422    -1.10452
0.09611
```

(3) We should create a MOLDEN input file. In this file, we treat all vectors as the normal-mode vectors.

```
 [Molden Format]
 [Atoms] Angs
```

```
N  1  7   -0.00303    -0.00061     0.10274
C  2  6    1.32564    -0.01016     0.13029
C  3  6    1.86507     1.36667    -0.00498
N  4  7    0.73128     2.16516    -0.11232
C  5  6   -0.37190     1.30725    -0.04334
N  6  7    3.13651     1.69280    -0.01421
C  7  6    4.01112     0.64176     0.11472
N  8  7    3.67269    -0.66086     0.24611
C  9  6    2.36234    -1.00803     0.25678
N 10  7    2.03113    -2.31445     0.38866
H 11  1    5.08724     0.88753     0.11014
H 12  1    1.07335    -2.60556     0.40041
H 13  1    2.74035    -3.01509     0.47705
H 14  1    0.71809     3.16593    -0.21866
H 15  1   -1.40093     1.64814    -0.10006
 [FR-COORD]           15
N   -0.00572    -0.00116     0.19384
C    2.50120    -0.01918     0.24582
C    3.51901     2.57863    -0.00939
N    1.37977     4.08520    -0.21193
C   -0.70170     2.46651    -0.08178
N    5.91795     3.19396    -0.02681
C    7.56814     1.21087     0.21645
N    6.92960    -1.24690     0.46436
C    4.45724    -1.90194     0.48449
N    3.83232    -4.36689     0.73333
H    9.59856     1.67459     0.20781
H    2.02518    -4.91615     0.75550
H    5.17046    -5.68886     0.90010
H    1.35489     5.97346    -0.41256
H   -2.64327     3.10970    -0.18879
 [FREQ]    10
        1000
        2000
        3000
 [FR-NORM-COORD]    10
Vibration            1
  0.04725    -0.02630     0.00373
 -0.06799    -0.00642    -0.00067
  0.04265     0.03102    -0.00244
  0.00961     0.02102    -0.00204
 -0.01244     0.01160    -0.00148
 -0.06270     0.01506    -0.00285
  0.04362    -0.00737     0.00165
 -0.00066    -0.02366     0.00250
  0.01897    -0.01509     0.00198
  0.00106     0.01120    -0.00117
 -0.00690    -0.00149     0.00002
 -0.00405    -0.00073     0.00000
  0.00346    -0.00349     0.00044
  0.00268    -0.01190     0.00132
 -0.01455     0.00655    -0.00098
Vibration            2
 -0.01106    -0.01304     0.00178
 -0.00622     0.00836    -0.00115
  0.02334    -0.02721     0.00235
 -0.00358     0.02292    -0.00193
  0.03056    -0.00623     0.00060
```

```
  0.00050     0.03311    -0.00259
 -0.03843     0.00118    -0.00184
  0.04479    -0.02768     0.00478
 -0.02682     0.00021    -0.00051
  0.01093     0.01739    -0.00175
 -0.00362    -0.00066    -0.00006
 -0.00491    -0.00061     0.00003
  0.00368    -0.00464     0.00033
  0.00215    -0.01283     0.00136
 -0.02131     0.00974    -0.00140
Vibration            3
  0.00107     0.00118    -0.00072
  0.00059    -0.00078     0.00024
 -0.00224     0.00269     0.00066
  0.00035    -0.00224    -0.00033
 -0.00293     0.00066     0.00054
 -0.00003    -0.00324    -0.00056
  0.00365    -0.00002     0.00102
 -0.00425     0.00255    -0.00130
  0.00256    -0.00002     0.00001
 -0.00104    -0.00165     0.00028
  0.00035     0.00007     0.00006
  0.00047     0.00005    -0.00006
 -0.00035     0.00047     0.00018
 -0.00020     0.00123    -0.00009
  0.00203    -0.00093     0.00008
```

In this file, I define three vibrational modes with random frequencies and treat the gradient vectors as the normal-mode vectors. Then I can view all gradient vectors by MOLDEN or MOLEKEL.

The first part of this file

```
 [Molden Format]
 [Atoms] Angs
```

defines the molecular structure in Angstrom.

The second part

```
[FR-COORD]           15
```

defines the molecular structure in Bohr (a.u.) which is used as the reference to define the displacement vectors.

The third part

```
[FREQ]    10
        1000
        2000
        3000
```

defines three vibrational modes with frequencies 1000cm-1, 2000cm-1 and 3000cm-1. They are used only by label all vectors.

The fourth part

```
[FR-NORM-COORD]    10
Vibration            1
...
Vibration            2
...
Vibration            3
...
```

defines the displacement vectors. Of course, the unit here should be atomic units. But in many situations, the length of these vectors are very small or very large. It is necessary to rescale them to obtain a better view.

After this preparation, MOLDEN (MOLEKEL) will treat this input as a input file containing the normal-model displacements. Then just open it and have a fun!

Sometime it is very important to write a MOLDEN file with the correct format. For more details of the MOLDEN format, please see the website

http://www.cmbi.ru.nl/molden/

If MOLDEN can not provide the pretty figures, try to use MOLEKEL because it also reads the MOLDEN input. For more details of MOLEKEL, please see the website

http://www.cscs.ch/index.php?option=com_content&task=view&id=90&Itemid=113

Dynamics calculations

From Thielopedia

Jump to: navigation, search

The investigation of the potential-energy surface of molecular excited states can provide us the general pictures of the nonadiabatic decay dynamics. If we wish to understand more insight, the dynamics simulations are necessary.

(1) The dynamics simulations should be performed by following the below procedure.

```
  (a) Sampling the initial conditions
  (b) On-the-fly surface-hopping dynamics
  (c) Average over a large number of trajectories.
```

(2) The initial sampling in fact composes two steps.

```
  (a) A large number of geometries close to the ground-state minimum should be
obtained.
  (b) According to the oscillator strength of each geometry, judge whether it can
be used
      for the initial condition of the nonadiabatic dynamics
```

(3) In general there are two ways to generate a large number of geometries near the ground-state minimum.

```
  (a)  The normal-mode of a molecular system is calculated. Then assume that each
mode is in its
vibrational ground level at zero temperature. Since each normal mode can be
described as an
independent harmonic oscillator, the nuclear wavefunction of the ground level of
each vibration
is a Gaussian functions. Then, the Wigner distribution function can be constructed
to describe the
distribution probability in the phase space. All geometries and their velocities
can be generated
according to this Wigner distribution function. From the mathematical point of
view, this approach
is not difficult in the normal-mode coordinate space. Since the wavefunction of
each mode here is
one-dimensional Gaussian, the corresponding Wigner distribution function is a two-
dimensional Gaussian
function.
  (b) The ground-state MD could be run to extract many snapshots.  In principle,
two steps should be
considered: equilibrium run and production run. In the first running, the system
tries to go to equilibrium.
In the second step, a large number of the snapshots should be extracted.
```

(4) The next question is how to choose the initial geometries.

```
   (1) Since we start from some excited states, we should first judge whether the
vertical excitation energy
      is correct for each geometry. In general, when the vertical excitation
energy of a geometry is close
      to its value at the equilibrium geometry, the electronic excitation is
possible.
   (2) Each geometry has its oscillator strength, which can also be used to judge
whether the electronic
       excitation is possible.
```

(5) Then the nonadiabatic dynamics should be calculated. In MNDO, the potential energies, all gradient vectors are calculated on-the-fly. The surface-hopping method is taken to treat the nonadiabatic transition near conical intersections.

(6) Finally, the result should be obtained by averaging a large number of trajectories.

For the initial samplings and data analysis, a few python scripts are useful, which could be found in the subdirectory of mdtools under directory of mndo99. The general introduction of these script is given in

Non-adiabatic dynamics with MNDO.

Ground-state MD simulation

From Thielopedia

Jump to: navigation, search

The below gives an example of ground-state MD running.

```
iop=-6 jop=-2 igeom=1 iform=1  +
icuts=-1 icutg=-1 iscf=9 iplscf=9 dstep=0.00001 kitscf=500 +
iprint=1 mprint=1 iprec=100 nprint=-1 +
imult=0 ioutci=1 +
icross=6, ipubo=1 ktrial=11 iroot=1 ncigrd=1 +
kci=5  imomap=1  +
movo=1 ici1=5 ici2=4 nciref=3 mciref=0 levexc=2 iroot=8

Guanine OM2
               6                  -0.00432  1           0.02401     1
0.00020   1
               6                   1.44324  1           0.00010     1
0.00256   1
               6                   2.10080  1          -1.27806     1
0.00159   1
               7                   1.52724  1          -2.51421     1
-0.00135   1
               6                   0.18938  1          -2.46014     1
-0.00348   1
               7                  -0.55711  1          -1.29072     1
-0.00285   1
               7                   2.38009  1           0.99778     1
0.00572   1
               6                   3.56970  1           0.39250     1
0.00672   1
               7                   3.44318  1          -1.00108     1
0.00427   1
               7                  -0.50520  1          -3.63766     1
-0.00649   1
               8                  -0.78959  1           0.97903     1
0.00043   1
               1                   4.53370  1           0.89870     1
0.00914   1
               1                   4.18585  1          -1.68214     1
0.00440   1
               1                  -1.57156  1          -1.35039     1
-0.00463   1
               1                  -1.49747  1          -3.67589     1
-0.00821   1
               1                  -0.01099  1          -4.50008     1
-0.00698   1

  24 25  26 27 28 29 30 31 32
  1
```

This example is used for the ground-state MD simulation.

```
 (1) "iscf=9 iplscf=9" should be for most dynamics simulations.
     With  "iscf>9 iplscf>9", the calculation is very easy to crash.
     With  "iscf<9 iplscf<9", the results are not very precise.
     Please do not set these two values smaller than 7 in all cases.
 (2) "imomap=1" is used to keep the active space containing the same (or similar)
```

```
orbitals.
 (3) "icorss=6" actives the MD simulations.
```

The ground-state MD running also needs the second input file called "dynvar.in"

```
&DYNVAR
IOUT    =           0,
NSTEP   =          50000,
DT      =  1.000000000000000E-004,
TEMP0   =   300.000000000000     ,
NOROT   = T,
NOTRA   = T,
INIT_STAT       =     1,
WRITE_STATS     = T,
SUNIT   =          30,
SFILE   = stat.out            ,
WRITE_TRAJ      = T,
XUNIT   =          31,
XFILE   = traj.out            ,
WRITE_VEL       = T,
VUNIT   =          32,
VFILE   = vel.out             ,
ENE_TOL =  1.000000000000000E-002,
FSTAT   =           1,
AVSTAT  = F,
FSAV    =           1,
RESTART = F,
WRITE_REST      = T,
RFILE   = dynam.restart       ,
EHRENFEST       = F,
TULLY_HOP       = F,
SIMHOP  =           0,
EINTEG  = UP3  ,
NE      =         100,
NUM_CC  = F,
AN_CC   = T,
RND_GEN = standard,
RND_SEED        =          -1,
WRITE_HOP       = F,
HOPUNIT =          33,
HOPFILE = hopping.out         ,
FHOP    =           1,
VS      = T,
TFIX    =   300.000000000000     ,
FVS     =           20,
VS_EMAX =  -1
/
```

This is the additional input. For the meaning of the keywords, please see below:

```
     iout         Int      0              Print level (0-3)
     nstep        Int      10             Number of MD steps
     dt           Double   5.0D-5         Time step (in picoseconds)
     temp0        Double   300.0D0        Initial temperature (K)
     norot        Logical  TRUE           Flag to adjust initial velocities to
```

```
remove
                                        overall angular momentum of the system
     notra        Logical  TRUE          Flag to adjust initial velocities to
remove
                                        overall linear momentum of the system
     init_stat    Int      1             Initial state
     write_stats  Logical  TRUE          Flag for writing of statistics
     sunit        Int      30            Fortran unit for statistics file
     sfile        Char(20) stat.out      File name for statistics
     write_traj   Logical  TRUE          Flag for writing of trajectory
     xunit        Int      31            Fortran unit for trajectory file
     xfile        Char(20) traj.out      File name for trajectory
     write_vel    Logical  FALSE         Flag for writing of velocities
     vunit        Int      32            Fortran unit for velocity file
     vfile        Char(20) traj.out      File name for velocities
     ene_tol      Double   1.0D-2        Obsolete option
     fstat        Int      1             Frequency for saving statistics to file
                                          = 1  every step
                                          = n  every n steps
     avstat       Logical  See text      Flag for turning on statistics averaging
                                        over fstat = n steps
                                          If n > 1, the default is TRUE, else
                                          FALSE
     fsav         Int      1             Frequency for saving trajectory and
                                        velocities to file
                                          = 1  every step
                                          = n  every n steps
     restart      Logical  FALSE         Flag to indicate if restarting dynamics
                                        from a restart file
     write_rest   Logical  TRUE          Flag for writing a restart file
     rfile        Char(20) dynam.restart Restart file name
     ehrenfest    Logical  FALSE         Flag to enable Ehrenfest dynamics
     tully_hop    Logical  FALSE         Flag to enable Tully surface hopping
     simhop       Int      0             Flag to enable simple hopping
                                          = 0   No simple hopping
                                          = 1   diabatic surface hopping
                                          = 2   diabatic surface hopping with
                                                average value of coupling (AS1)
                                          = 3   average coupling compared with
                                                random number
                                          = 4   average coupling compared with
                                                random number + velocity
                                                adjustment (AS2)
                                          = 5   average coupling compared with
                                                random number, hop only with
                                                negative coupling + velocity
                                                adjustment
                                          = 6   Average coupling correction;
                                                average coupling compared with
                                                random number + velocity
                                                adjustment (AS3)
                                         (note ehrenfest,tully_hop and simhop
                                          are mutually exclusive)
     einteg       Char(5)  UP3           Integration algorithm for use with the
                                        Tully electronic equation
                                           UP1  = Unitary exponential propagator
                                                  computed at midpoint.
Eigenvalue
```

			expansion used to build the exponetial UP2 = Unitary exponential propagator computed at the beginning of the interval. Eigenvalue expansion used to build the exponetial UP3 = Unitary exponential propagator computed at midpoint. Taylor expansion used to build the exponetial ABM5 = Adams-Bashforth-Moulton 5-th order predictor-corrector with local truncation error evaluation. Also iterative SCF cycles are performed if error is greater than 1.D-10 ABM4 = Adams-Bashforth-Moulton 4-th order predictor-corrector with local truncation error evaluation. Also iterative SCF cycles are performed if error is greater than 1.D-10 ABM2 = Adams-Bashforth-Moulton 5-th order predictor-corrector RK2 = second order Runge-Kutta RK4 = fourth order Runge-Kutta euler = Euler method (only for testing purposes)
ne	Int	200	Number of steps for integration of the Tully electronic equation
num_cc	Logical	FALSE	Flag for calculation of numerical non-adiabatic couplings
an_cc	Logical	TRUE	Flag for calculation of analytical non-adiabatic couplings (num_cc and an_cc are not mutually exclusive, but if both are selected only the analytical couplings are used in the Tully procedure. If numerical couplings only are selected, analytical couplings must still be calculated each time hopping occurs)
rnd_gen	Char(8)	standard	Algorithm for generating random numbers for initial velocity assignment and the Tully surface hopping algorithm standard = intrinsic Fortran subroutine (RANDOM_NUMBER). NB: this algorithm is compiler-dependent

```
                                                  PM_BD    = Park and Miller algorithm
with
                                                            Bays-Durham shuffle
                                                  knuth    = Knuth subtractive algorithm
     rnd_seed     Int      -1             Integer to seed the random number
generator
                                                  -1 = random seed (e.g. based on the
                                                       system clock)
                                                  NB: For rnd_gen = standard, each
member
                                                      of the seed array is set to
rnd_seed
     fol_stat     Logical  TRUE           Flag to enable state following
     write_hop    Logical  See text       Flag for writing of hopping data
                                                  Default is TRUE if Ehrenfest or Tully
                                                  dynamics are selected, FALSE otherwise
     hopunit      Int      33             Fortran unit for hopping data file
     hopfile      Char(20) hopping.out    File name for hopping data
     fhop         Int      1              Frequency for saving hopping data to
file
                                                  = 1  every step
                                                  = n  every n steps
     vs           Logical  FALSE          Flag for enabling velocity scaling
     tfix         Double   [=temp0]       Target temperature for velocity scaling
     fvs          Int      1              Frequency for performing velocity
scaling
                                                  = 1  every step
                                                  = n  every n steps
     vs_emax      Double   -1.0D0         Maximum kinetic energy correction
allowed
                                                  during velocity scaling
                                                  NB: a negative value implies no limit
```

In the equilibrium step, some keywords are important.

The temperature during the MD running should be check to see whether the results are reasonable. Therefore, you should save the statistics file.

```
WRITE_STATS     = T,
SUNIT   =          30,
SFILE   = stat.out ,
```

The MD should start only from the input geometry. But you need to save the final geometry and velocity for the next production running.

```
RESTART = F,
WRITE_REST      = T,
RFILE   = dynam.restart       ,
```

Of course, this is the ground-state MD. Therefore we have

```
EHRENFEST       = F,
TULLY_HOP       = F,
SIMHOP  =           0,
```

I generally recommend the analytical gradient because it is precise and calculation of this analytical gradient is faster

```
AN_CC   = T,
```

During the MD running, the temperature of the systems may go wrong. Therefore, it is necessary to fix the temperature. In order to do it, the rescaling of the velocity is applied for this purpose. As the equilibrium running, we should rescale the velocity quite frequency and permit the large change.

```
VS      = T,
TFIX    =   300.000000000000     ,
FVS     =           20,
VS_EMAX =  -1
/
```

After the equilibrium, the production running is performed to have all geometries.

(1) The same MNDO input files could be used.

(2) Some keywords in the file "dynvar.in" should be changed.

```
    RESTART = T,
    WRITE_REST      = F,
    RFILE   = dynam.restart
```

In this situation, the restart file "dynam.restart" that contains the final geometry and velocity will be taken for the initial condition of the production running. Please note that the geometry in the MNDO input file is not useful although you specify it. Since it is not necessary to write the final geometry and velocity at the end of the production running,

```
WRITE_REST      = F,
```

is set to avoid the overwrite the "dynam.restart".

(3) To avoid the disturb of the system dynamics in the production running, the velocity rescaling should be applied quite often and largely. For instance

```
VS      = T,
TFIX    =   300.000000000000     ,
FVS     =           200,
VS_EMAX =  0.2
/
```

Initial samplings

From Thielopedia

Jump to: navigation, search

After the generation of a large number of the geometries, the next question is to judge whether these geometries can be excited to the electronic excited states. As mentioned in Dynamics calculations, a few criterion should be considered here.

```
  (a) A reference geometry should be set. The molecular orbitals, the vertical
excitation energy of this reference
      geometry should be calculated. In general, this reference geometry should be
the equilibrium geometry of the
      ground state.
  (b) If the initial excitations bring the system to a particular state, the
energy window is added to reject the
      geometries with the vertical excitation energy out of the this window. This
means only the excitation energy
      is close the reference energy, this geometry can be accepted.
  (c) The transition probability is related to the oscillator strength. A
stochastic method is employed to judge
      whether a geometry can be accepted or rejected. The relative probability t_i
      t_i = f_i/max(f_i)
      is defined as the oscillator strength (f_i) of Geometry i divided by the
maximum oscillator strength among all
      geometries. In this case, t_i is between [0,1]. A set of random numbers p_i
are generated between [0,1].
      If p_i < t_i, Geometry i is accepted.
      If p_i > t_i, Geometry i is rejected.
  (d) The active space for all trajectories should have the same orbitals.
Therefore, the molecular-orbital mapping
      methods is applied. If overlap of the molecular orbital is too small. This
means that this geometry is far
      way from the reference geometry. Reject it.
```

(1) In many situations, (a) does not work for the large system. (b) is not very important. (c) is the main judgement.

(2) For the initial sampling, a few python scripts

```
mdsample.py
mdfilter.py
```

are useful. Before running these scripts, make sure that you always include

```
"mndotools.py"
```

because it is a subroutine used for almost all other scripts.

(3) After the ground state MD production running, two files

```
    traj.out
    vel.out
```

are used as the input for the initial sampling.

In many cases, both of these two files contains many geometries. Not all of them are needed for the selections. Therefore, we first random select 200-400 geometries from these two output files. Here,

```
mdsample.py
```

is used for this purpose and its parameters are given below

```
-h, --help
       display this help message
-q, --quiet
        don't print anything to standard output
--trajectory=(file name)
       set a new input file name
--sample=(file name)
       set a new output file name
--log=(file name)
       set a new log file name
--sample-size=(int)
       set the number of sample geometries required
       (default: as many as possible)
--random-seed=(int)
       set the seed for the random number generator
--step=(int)
       set a fixed interval for sampling instead of taking a random sample
       A step of 1 means that all geometries will be sampled.
--step-start=(int)
       set the first geometry to be sampled in the case of fixed step
       sampling, numbered from 0. (default: step size - 1)
```

For instance, if we wish to have 200 geometries and their velocities, the following command

```
./mdsample --sample-size=200 --vel
```

is used. Then two output files

```
sample.xyz
sample.vel
```

contain the 200 geometries and their corresponding velocities, respectively.

If we wish to have 300 geometries without their velocities, the following command is used

```
./mdsample --sample-size=200
```

is used. Then only one output file

```
sample.xyz
```

is generated.

(4) In the following step, the real initial sampling is performed to select geometries and velocities from

```
sample.xyz
sample.vel
```

based on the criterion (a), (b), (c) and (d). The script

```
mdfilter.py
```

is used for this purpose. Here is the instruction

```
"""mdfilter - filter a list of sample geometries
PURPOSE
mdfilter takes a list of sample geometries in XYZ format and filters them
according to a number of criteria:
1. Successful mapping of active space
    - Each geometry is tested against a reference (template)
      (where the active space is assumed to be known for certain).
      If the active space cannot be mapped with a confidence over
      a certain threshold (default: 90%), it is rejected.
      Active space mapping can be switched off with the option --no-map
2. Energy window
    - The excitation energy of each state is compared to that
      of the reference (or a user defined value). If it does not lie
      within a given tolerance (default: 0.15eV), it is rejected.
      This filtering can be surpressed with the option --no-window
3. Transition probability
    - The transition probability is calculated (following Ref. Eq. 4) as
                P = (f/DE^2) / max(f/DE^2)
      where f is the oscillator strength, and DE is the energy gap
      between states. For each state, a random number 0 < n < 1 is
      generated, and if n > P, the state is rejected.
      This filtering can be surpressed with the option --no-prob
      Ref: M. Barbatti et al., J. Photochem. Photobiol. A, v190, p228 (2007)
In addition, filters (2) and (3) are limited by default to excitations
to a single target excited state (default: 2). This restriction can be
removed with the option --all-states
The default output is a list of XYZ geometries, with details in the title
line of the selected state and mapped active orbitals.
Additionally, MNDO input files can be generated for subsequent excited state
dynamics runs (using the option --dynamics), and/or an absorption
spectrum can be simulated (using the option --spectrum)
The filtered geometries can be used as the basis of an excited states
dynamics run or simply to simulate an absorption spectrum.

 REQUIREMENTS
Requires the mndotools.py library



USAGE
mdfilter.py [OPTIONS]



GENERAL INPUTS (default file names)
template.inp - MNDO template input file, detailing CI options,
               reference geometry and active space, etc.
               Usually would be a ground state optimised geometry or
               equilibrated geometry of some kind.
sample.xyz   - List of sampled geometries in XYZ format

INPUT FOR DYNAMICS PREPARATION (option --dynamics)
dynvar.in    - Template file for dynamics input
               (this will be left unchanged except possibly for
                init_stat and restart)

INPUT FOR REFILTERING (option --refilter)
```

```
filtered-info.log       - Output from a previous mdfilter run. Use the raw
                          data from this log file instead of recalculating
                          the data with an MNDO calculation. The refiltering
                          input must have been produced using the same
                          template molecule and sample XYZ geometries.

GENERAL OUTPUTS (default file names)
(standard output)       - Basic progress information
filtered-sample.xyz     - List of geometries that passed through the filter.
filtered-info.log       - Summary information and further information for
                          each geometry
                          (active space, excitation energy, probability).

OUTPUT FOR DYNAMICS PREPARATION (option --dynamics)
For each trajectory (here the first),
run0001/mndo.inp        - Input file with geometry and standard MNDO options
                          (based on template.inp)
run0001/dynvar.in       - Input file with dynamics options
                          (unchanged except for init_stat/restart)

OUTPUT FOR SPECTRUM SIMULATION (option --spectrum)
filtered-spectrum.dat   - Histogram data (default bin width=0.1eV) for
                          simulated absorption spectra.
DEFAULTS
General calculation options
  - Path to MNDO executable: the output of 'which mndo-md'
  - Target excited state: 2
  - Oscillator strength formalism: rp
Active space filtering
  - Threshold for successful mapping: 0.9
Energy window filtering
  - Target excitation energy: that of the template file (state 2)
  - Tolerance for energy window filtering: 0.15eV
Dynamics preparation
  - ncigrd gradients: from ground state up to target-state
                      (or up to iroot with --all-states)
Absorption spectrum
  - Histogram bin size for absorption spectrum: 0.1eV
All defaults can be changed using the command line options below.

GENERAL OPTIONS
-h, --help
       display this help message
-q, --quiet
       don't print anything to standard output
--mndo-path=(path string)
       specify a new path for MNDO. Default is the result of 'which mndo-md'
-r, --refilter
       do not rerun sample MNDO calculations, but read quantities directly
       from the mdfilter log file from a previous run (must correspond to the
       same input template and xyz files)

OPTIONS FOR STANDARD INPUT FILE NAMES
--template=(file name)
       specify a different file name for the template molecule
--sample=(file name)
       specify a different file name for the sample geometry list
```

```
OPTIONS FOR FILTERING BY STATE
--target-state=(int)
       specify a new target excited state (integer).
       By default all states are rejected except the target state
       This option is also used by energy window filtering (see below)
-a, --all-states
       consider possibility of excitation to any state
       (i.e. do not filter by state)

OPTIONS FOR FILTERING BY ACTIVE SPACE MAPPING
--mapthr=(float)
       specify a new threshold for filtering by active orbital mapping (0-1)
       This option is also used when preparing dynamics inputs (see below)
-M, --no-map
       do not attempt to map the active molecular orbitals
       (i.e. imomap=0). With this option values of movo != 1 can be used.

OPTIONS FOR FILTERING BY ENERGY WINDOW
--target-state=(int)
       specify a new target excited state (integer)
       The excitation energy from the template molecule for the given state
       will be used as the midpoint of the energy window.
--window-target=(real)
       specify a target excitation energy for the energy window
       (instead of taking the value from the template molecule calculation)
--window-tolerance=(real)
       specify a new tolerance for energy window filtering
-W, --no-window
       do not reject structures whose excitation energies fall outside the
       energy window.

OPTIONS FOR FILTERING BY TRANSITION PROBABILITY
--f-formalism=r|p|rp
       MNDO provides oscillator strengths calculated using three different
       formalisms (which are identical only in the limit of an exact
       wavefunction).
       This option specifies which formalism should be used in calculating
       the transition probability.
       (note: cannot be changed when using --refilter)
--random-seed=(int)
       specify a seed for the random numbers used by the transition
       probability filter
-P, --no-prob
       do not reject structures which have been rejected according to the
       stochastic algorithm based on transition probability.

OPTIONS FOR REFILTERING
-r, --refilter
       do not rerun sample MNDO calculations, but read quantities directly
       from the mdfilter log file from a previous run (must correspond to the
       same input template and xyz files)
--refilter-info=(file name)
       specify a new name for the refilter info input file

OPTIONS FOR STANDARD OUTPUT FILE NAMES
--filtered-info=(file name)
       specify a new name for the log file
```

```
--filtered-sample=(file name)
       specify a new name for the filtered list of sample geometries

OPTIONS FOR PREPARATION OF DYNAMICS RUN FILES
-d, --dynamics
       prepare excited state dynamics run files
--md-states=(string)
       specify the gradients to be calculated during the dynamics runs.
       By default, gradients will be calculated for the ground state up to
       the target state (or 'iroot' in the case of --all-states).
       With this option a specific set of states can be given
       (separated by spaces, e.g. --md-states='1 2 3' for the first 3 states)
       Note this may result in dynamics runs that crash, if init_stat is
       not included in --md-states.
--mapthr=(float)
       specify a new threshold for active orbital mapping during the dynamics
       run (0-1, converted to a percentage in the MNDO input)
       This option is also used for active space filtering (see above)
-M, --no-map
       if filtering by active space mapping has been switched off,
       mapping will also be off during the dynamics run (imomap=0)
--dynvar=(file name)
       specify a new file name for the dynamics options input file
--md-prefix=(string)
       specify a new prefix for the run file directories
--vel
       provide initial velocities for dynamics runs from a sample velocity
       file. In this case a minimal restart file will be created for each
       run containing the appropriate initial velocities.
--vel-inp=(file name)
       specify a new file name for the sample velocity input file

OPTIONS FOR SIMULATION OF ABSORPTION SPECTRA
-s, --spectrum
       write an absorption spectrum file
--bin-size=(float)
       specify a new histogram bin size for the absorption spectrum (in eV)
--filtered-spectrum=(file name)
       specify a new name for the histogram absorption spectrum file
```

This script is not so easy to use. Therefore, when you need to use it, please always ask the person who knows how to run this script.

In the initial sample, you also need to have a template file "template.inp" containing the reference geometry and some MNDO keywords. For instance

```
iop=-6 igeom=1 iform=1 icuts=-1 icutg=-1 +
iscf=9 iplscf=9 dstep=0.00001 MAXRTL=500 kitscf=500 +
imult=1 ioutci=1 +
kci=5    +
movo=1 ici1=6 ici2=3  nciref=3 mciref=0 levexc=2 iroot=3 +
mprint=-1 nprint=-5

OM2
 6            -0.0057380957       1       0.0257873395       1       0.0002140962
```

```
1
 6            1.4447679693       1       0.0020683292       1       0.0025724681
1
 6            2.1021705608       1      -1.2795562009       1       0.0015839991
1
 7            1.5248913604       1      -2.5170068194       1      -0.0013577319
1
 6            0.1917275039       1      -2.4597054588       1      -0.0034690203
1
 7           -0.5561632917       1      -1.2907123937       1      -0.0028182200
1
 7            2.3827316163       1       0.9923625639       1       0.0057173267
1
 6            3.5672607430       1       0.3908173981       1       0.0066969675
1
 7            3.4461876874       1      -1.0023681198       1       0.0042583442
1
 7           -0.5065173443       1      -3.6386666714       1      -0.0065247503
1
 8           -0.7941359802       1       0.9789942107       1       0.0004447582
1
 1            4.5244955467       1       0.8954821324       1       0.0090933421
1
 1            4.1917754731       1      -1.6810907208       1       0.0043826698
1
 1           -1.5707946077       1      -1.3528849525       1      -0.0045979799
1
 1           -1.5001314076       1      -3.6753932559       1      -0.0082235986
1
 1           -0.0133991694       1      -4.5028402921       1      -0.0071012447
1

   24 25  26 27 28 29 30 31 32
```

"mdfilter.py" can also be used to generated the input files. Therefore, before running this script, you also need to have "dynvar.in".

```
&DYNVAR
IOUT    =          1,
NSTEP   =       16000,
DT      =  1.000000000000000E-004,
TEMP0   =   300.00000000000     ,
NOROT   = T,
NOTRA   = T,
WRITE_STATS     = T,
SUNIT   =         30,
SFILE   = stat.out           ,
WRITE_TRAJ      = T,
XUNIT   =         31,
XFILE   = traj.out           ,
WRITE_VEL       = F,
VUNIT   =         32,
VFILE   = vel.out            ,
FSTAT   =          1,
AVSTAT  = F,
FSAV    =          10,
RESTART = T,
WRITE_REST      = F,
```

```
RFILE   = dynam.restart       ,
EHRENFEST       = F,
TULLY_HOP       = T,
EINTEG  =  UP3 ,
NE      =        200,
NUM_CC  = F,
AN_CC   = T,
RND_GEN = PM_BD,
WRITE_HOP       = T,
HOPUNIT =          33,
HOPFILE = hopping.out         ,
FHOP    =           1,
VS      = F,
/
```

One additional question which should be considered here is whether we need to include the velocity when we prepare the input file for the excited-state dynamics. In general, I always recommend that you include them if possible. However, in some cases, the ground state MD running is not stable. If we wish to obtain enough geometries, we have to rescale the velocity quite often and use very low temperature. Then the velocity is not very reliable and we should not include them in the next nonadiabatic surface-hopping simulations.

On-the-fly surface-hopping calculations of nonadiabatic dynamics at conical intersections

From Thielopedia

Jump to: navigation, search

After generation of all input files by runnning "mdfilter.py", we will have many directories named as

```
./run0001
./run0002
...
```

Each directory contains a input file for the excited-state dynamics.

For instance, "mndo.inp" file is the main file to be used as the input.

```
iop=-6 igeom=1 iform=1 icuts=-1 icutg=-1 +
iscf=9 iplscf=9 dstep=0.00001 maxrtl=500 kitscf=500 +
imult=1 ioutci=1 +
kci=5 +
movo=1 ici1=6 ici2=3 nciref=3 mciref=0 levexc=2 iroot=3 +
mprint=-1 nprint=-5 +
jop=-2 ncigrd=3 ktrial=11 icross=6 ncisym=-1 imomap=1 ipubo=1
Geom 1, State 2
```

```
Generated automatically by mdfilter.py
 6         -0.0031098300  0          0.0349808900  0         -0.0705302600  0
 6          1.4425350300  0         -0.0078438800  0         -0.1870691800  0
 6          2.1417710700  0         -1.2582346600  0          0.0257263000  0
 7          1.5246487000  0         -2.4852488400  0          0.0622254100  0
 6          0.2193256300  0         -2.4529143700  0         -0.0044997500  0
 7         -0.5879465400  0         -1.2863441300  0         -0.0997429200  0
 7          2.4345549100  0          0.9621709700  0         -0.1386478000  0
 6          3.5919182100  0          0.4004013100  0          0.0008050100  0
 7          3.4437362500  0         -1.0053278900  0          0.1791188400  0
 7         -0.5163828000  0         -3.5958564500  0         -0.0418007600  0
 8         -0.8727596800  0          0.9129710500  0          0.1814898000  0
 1          4.5543368700  0          0.9385450600  0          0.0087111000  0
 1          4.1449369700  0         -1.6395780200  0          0.5316108500  0
 1         -1.4959837200  0         -1.3827130900  0          0.3116062600  0
 1         -1.4736171200  0         -3.6879806900  0         -0.2158058700  0
 1         -0.0297844600  0         -4.4321264600  0         -0.0906819000  0
 0          0.0           0          0.0           0          0.0           0
24 25 26 27 28 29 30 31 32
1 2 3
```

For this input, there is nothing new.

```
 "ktrial=11"  generates the file fort.10 to save the temperature data, such as the
molecular orbitals.
 "icross=6"   Dynamics.
```

Of course, the file "dynvar.in" is needed

```
&DYNVAR
IOUT = 1,
NSTEP = 16000,
DT = 1.0000000000000000E-004,
TEMP0 = 300.00000000000,
NOROT = T,
NOTRA = T,
WRITE_STATS = T,
SUNIT = 30,
SFILE = stat.out,
WRITE_TRAJ = T,
XUNIT = 31,
XFILE = traj.out,
WRITE_VEL = F,
VUNIT = 32,
VFILE = vel.out,
FSTAT = 1,
AVSTAT = F,
FSAV = 10,
RESTART = T,
WRITE_REST = F,
RFILE = dynam.restart,
EHRENFEST = F,
TULLY_HOP = T,
EINTEG = UP3,
NE = 200,
NUM_CC = F,
AN_CC = T,
```

```
RND_GEN = PM_BD,
WRITE_HOP = T,
HOPUNIT = 33,
HOPFILE = hopping.out,
FHOP = 1,
VS = F,
INIT_STAT = 2,
/
```

Please notice

```
  INIT_STAT = 2 : The initial state is the S1 state.
  VS = F        : We can not do the velocity rescaling during the simulation
because this is the surface-hopping study.
  TULLY_HOP = T : Tully's method is used for the surface-hopping calculations.
  RESTART = T,  : We start from restart file.
                  If  RESTART = F, we start from only the MNDO input geometry.
```

Here is an example of the restart file.

```
#restart   MD_only
#N_atoms    16
#coordinates
        -0.0031098300         0.0349808900        -0.0705302600
         1.4425350300        -0.0078438800        -0.1870691800
         2.1417710700        -1.2582346600         0.0257263000
         1.5246487000        -2.4852488400         0.0622254100
         0.2193256300        -2.4529143700        -0.0044997500
        -0.5879465400        -1.2863441300        -0.0997429200
         2.4345549100         0.9621709700        -0.1386478000
         3.5919182100         0.4004013100         0.0008050100
         3.4437362500        -1.0053278900         0.1791188400
        -0.5163828000        -3.5958564500        -0.0418007600
        -0.8727596800         0.9129710500         0.1814898000
         4.5543368700         0.9385450600         0.0087111000
         4.1449369700        -1.6395780200         0.5316108500
        -1.4959837200        -1.3827130900         0.3116062600
        -1.4736171200        -3.6879806900        -0.2158058700
        -0.0297844600        -4.4321264600        -0.0906819000
#velocity
        -1.4479433700        -4.7152966900        -1.9461195100
        -2.6471382000         8.1465175500        -4.2179139700
        -1.9543498000        -1.1477215500        -4.3134094300
        -2.2296189900         2.7194464200         2.6848774700
         4.0974898400         0.1950959300         2.3964611600
        -2.9259151000        -1.2608194800         7.7258191800
         3.5645289000        -2.6915932500         1.3394848200
         1.4988606300        -3.1838916500        -2.9719978900
         0.0288269100         2.2176338700         6.5574700900
         1.2260442700        -0.2716646400        -6.6247946700
         1.9912168500         2.2084588900         0.0596705700
       -20.1361504100       -11.9279781900       -33.4033859600
        11.1456687100         0.9591113800        10.1972568000
       -21.4312426500       -27.4871960700         4.5293864900
         5.5026039900         1.7393603300       -16.7427343500
         3.3821907100         0.1579993800         3.7527360300
```

Data analysis

From Thielopedia

Jump to: navigation, search

After the surface-hopping calculations. Two output files are important:

```
hopping.out:  All useful information is written inside.
stat.out:     Many statistical information.
traj.out:     The trajectory file
```

A few scripts should be used to extract the useful information from trajectories. They are

```
MDextr.py
mmean.py
geoman.py
mndotools.py
```

Always remember to copy mndotools.py into your working directory.

(1) The first step is to extract the data of the potential energies, the nonadiabatic couplings from the output file. He is an example.

```
 In the directory "./run0001" containing "hopping.out", type the command
./MDextr.py
 A subdirectory ./MD_data will be created and it contains all useful information.
 E1.dat:  Time-dependent potential energy for State 1.
 E2.dat:  Time-dependent potential energy for State 2.
 E12.dat: Time-dependent potential energy difference between State 2 and State 1.
 ...
 CC12.dat: Time-dependent nonadiabatic couplings between State 2 and State 1.
 ...
```

The actual potential energy during the simulation can be found in

```
  Column 4 of  "traj.out"
```

(2) If you wish to plot the population decay vs time, the script "mmean.py" should be used. An input contains all files need to be averaged. For example, the time-dependent occupation number is saved in

```
 ./MD_data/state.dat
```

An input file named as "data.ls" should be created as below

```
./run0001/MD_data/state.dat
./run0002/MD_data/state.dat
./run0003/MD_data/state.dat
./run0004/MD_data/state.dat
./run0005/MD_data/state.dat
./run0006/MD_data/state.dat
```

```
./run0007/MD_data/state.dat
./run0008/MD_data/state.dat
./run0009/MD_data/state.dat
 ...
Please delete some line in "data.ls" because all data list to be averaged should
have the same length.
```

Then run ./mmean.py -s You will have the time-dependent populations.

(3) If you wish to plot the internal coordinates changing vs time, the script "geoman.py" should be used. The file

```
traj.out
```

is the input file. He is the instruction of this script

```
Routine for analysing trajectories.
# Parse command-line options
usage = 'usage: %prog [options] file'
description = 'Extracts XYZ geometries or geometrical parameters from ' + \
              'a Molden or XYZ trajectory file. By default, atoms and ' + \
              'geometries are numbered starting from one. Negative ' + \
              'numbers count back from the last atom/geometry.'
parser = optparse.OptionParser(usage=usage, description=description)
parser.set_defaults(param=None, range=[], atomList=[], scaleIndex=1,
                    countStart=1)
parser.add_option('--range', '-r', action='store', type='int', nargs=2,
                  dest='range', metavar='R1 R2',
                  help='limit output to geometry range from R1 to R2')
parser.add_option('--length', '-l', action='callback', type='int', nargs=2,
                  callback=callbackParam, dest='length', metavar='A1 A2',
                  help='calculate bond length (magnitude of A1-A2)')
parser.add_option('--angle', '-a', action='callback', type='int', nargs=3,
                  callback=callbackParam, dest='angle', metavar='A1 A2 A3',
                  help='calculate bond angle (angle between vectors '
                  'A1-A2 and A2-A3)')
parser.add_option('--dihedral', '-d', action='callback', type='int',
                  nargs=4, callback=callbackParam, dest='dihedral',
                  metavar='A1 A2 A3 A4',
                  help='calculate dihedral angle (angle between planes '
                  'A1-A2-A3 and A2-A3-A4)')
parser.add_option('--outofplane', '-o', action='callback', type='int',
                  nargs=4, callback=callbackParam, dest='outofplane',
                  metavar='A1 A2 A3 A4',
                  help='calculate out of plane angle (angle between '
                  'plane A1-A2-A3 and vector A2-A4)')
parser.add_option('--scale-index', '-s', action='store', type='int',
                  dest='scaleIndex', metavar='N',
                  help='scale output of step numbers by a factor N')
parser.add_option('--zero-based', '-z', action='store_const', const=0,
                  dest='countStart',
                  help='count atoms/geometries starting from zero instead '
                  'of one')
```

For example:

```
 ./geoman.py -l 1 3    ./traj.out >    ./dis_1_3.dat :          Time-dependent
distance 1_3
 ./geoman.py -a 2 5 10 ./traj.out >    ./angle_2_5_10.dat :     Time-dependent
angle 2_5_10
 ./geoman.py -d 1 2 5 10 ./traj.out >  ./dih_1_2_5_10.dat :     Time-dependent
dihedral angle 1_2_5_10
```

(4) We have to clarify an additional thing. In the surface-hopping calculations, the calculations are performed in the adiabatic representation. However, when the system gives the output, it try to follow the electronic characters. This means that when the electronic character remains unchanged the output show that the system staying on the same state. On the other hand, if the electronic characters change, the output show that the system switch the state. In this sense, the output gives the state more like the so-called "diabatic states" although the calculation in fact is doing in the adiabatic representations. This output in some sense is fine because it can follow the changing of the electronic characters and approximately give the information of the electronic population in "diabatic representations". The code is developed in this way because the diabatic representation can give us more informations about the state characters such as npi* or pipi*.

However, this approach does not give you the real diabatic populations. In some other situations, this may results the problematical results. If the system passes a region with the strong electronic mixing, the electronic characters may in fact change when leaving this region. However, such transformation of the electronic characters maybe quite smooth. One such example is that if the nuclear motion make a half-loop containing a conical intersection and system always stay on lower surface, the electronic characters should change quite smooth. But at the end, the character completed switch to another one. In this case the present treatment may can not see such changing of the electronic characters.

Based on these reasons, it is worthwhile to recover the real adiabatic states and their occupation numbers. This should be quite easy because the only thing should be done is to resort the data in "hopping.out" according to the energy order. Please ask other people in this group to understand this point more.