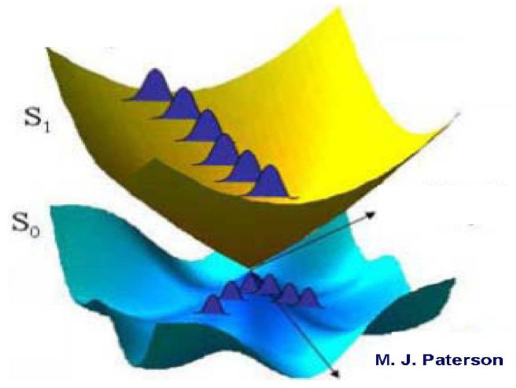
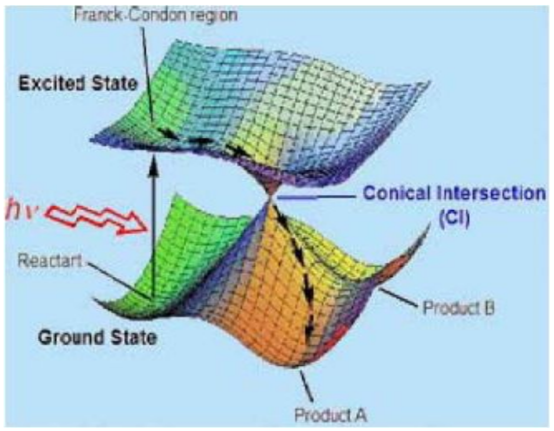
A brief introduction to JADE package

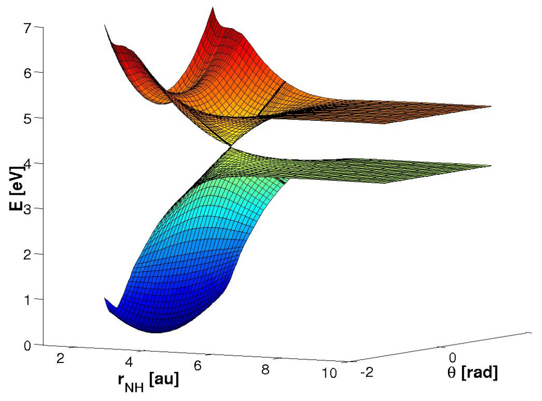
L. Du, Z. Lan*

https://github.com/jade-package

Nonadiabatic dynamics







Born-Oppenheimer approximation breaks down !!

A self-consistent treatment of nuclear and electronic degrees of freedom is needed.

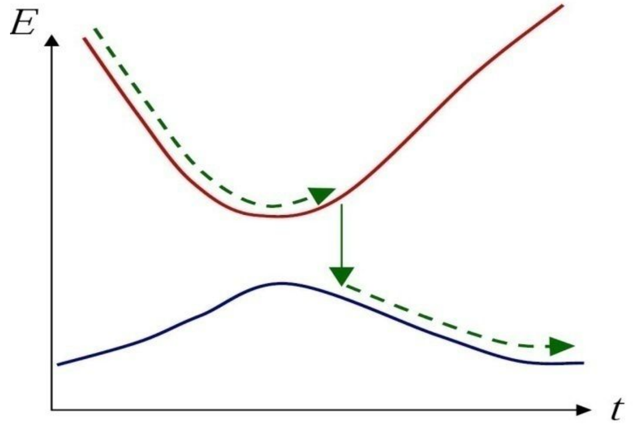
Surface hopping method

Methodology

A swarm of independent trajectories is considered.

Each trajectory evolves on a single potential energy surface.

Sudden hops between different potential energy surfaces are allowed.



Expansion of the electronic wavefunction

$$\Phi(\mathbf{r},\mathbf{R},t)=\sum_i c_i(t)\phi_i(\mathbf{r},\mathbf{R})$$

The electronic wavefunction is expanded as a superposition of independent adiabatic states.

Quantum amplitudes determine the time-dependent population of state i at time t

nonadiabatic coupling

Substitution into the time-dependent Schrödinger equation yields a set of coupled equations for the quantum amplitudes.

$$i\hbar\frac{dc_j(t)}{dt}=c_j(t)\epsilon_j-i\hbar\sum_i c_i(t)\dot{\mathbf{R}}\cdot\mathbf{d}_{ji}$$

J. C. Tully, *J. Chem. Phys.* **93**, 1061 (1990).

Tully's Surface Hopping Approach

$$i\hbar\frac{\partial\Phi(\mathbf{r},\mathbf{R},t)}{\partial t} = H_e\Phi(\mathbf{r},\mathbf{R},t).$$

$$\Phi(\mathbf{r},\mathbf{R},t) = \sum_i c_i(t)\phi_i(\mathbf{r},\mathbf{R}),$$

$$i\hbar\frac{dc_j(t)}{dt} = \sum_i c_i(t)[H_{ji} - i\hbar\dot{\mathbf{R}}\cdot\mathbf{d}_{ji}],$$

transition probability

$$P_{ij} = -\frac{2\int_t^{t+\Delta t} dt[\hbar^{-1}\mathrm{Im}(c_i^* c_j \epsilon_i \delta_{ji}) - \mathrm{Re}(c_i^* c_j \dot{\mathbf{R}}\cdot\mathbf{d}_{ji})]}{|c_i(t)|^2}.$$

$$H_{ji} \equiv \int d\mathbf{r}\phi_j^*(\mathbf{r},\mathbf{R}) \left[-\frac{\hbar}{2} \sum_l \frac{1}{m_l} \nabla_{\mathbf{r}_l}^2 + V_{rR}(\mathbf{r},\mathbf{R}) \right] \phi_i(\mathbf{r},\mathbf{R}),$$

$$\mathbf{d}_{ji} \equiv \int d\mathbf{r}\phi_j^*(\mathbf{r},\mathbf{R})[\nabla_{\mathbf{R}}\phi_i(\mathbf{r},\mathbf{R})].$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \mathbf{R}} \frac{d\mathbf{R}}{dt}$$

$$F_{ij}(t) = \int d\mathbf{r}\phi_j^*(\mathbf{r},\mathbf{R}) \frac{\partial \phi_i(\mathbf{r},\mathbf{R})}{\partial t}$$

$$\boxed{H_{ji} = \epsilon_i \delta_{ji}.}$$

$$i\hbar \frac{dc_j(t)}{dt} = c_j(t)\epsilon_j - i\hbar \sum_i c_i(t)\dot{\mathbf{R}} \cdot \mathbf{d}_{ji}.$$

$$P_{ij} = \frac{2\int_t^{t+\Delta t} dt \mathrm{Re}(c_i^* c_j \dot{\mathbf{R}} \cdot \mathbf{d}_{ji})}{|c_i(t)|^2}.$$

$$\rho_{kl} \equiv c_k c_l^*$$

$$\frac{d\rho_{ll}}{dt} = i\hbar^{-1}\left[\left(-H_{kl}^c + i\hbar \mathbf{F}_{kl}^c \cdot \mathbf{v}^c\right)\rho_{lk} - \left(-H_{lk}^c + i\hbar \mathbf{F}_{lk}^c \cdot \mathbf{v}^c\right)\rho_{kl}\right]$$
$$= -2\hbar^{-1} H_{kl}\,\mathrm{Im}\left(\rho_{kl}\right) - 2\,\mathbf{F}_{kl}^c \cdot \mathbf{v}^c\,\mathrm{Re}\left(\rho_{kl}\right).$$

$$P_{l\to k} = \max\left[0, \frac{-2\Delta t}{\rho_{ll}}\mathrm{Re}\left(\rho_{kl}\right)\mathbf{F}_{kl}^c \cdot \mathbf{v}^c\right]\text{(adiabatic)},$$

$$P_{l\to k} = \max\left[0, \frac{2\Delta t}{\hbar\rho_{ll}}\mathrm{Im}\left(\rho_{kl}\right)W_{lk}^c\right]\text{(diabatic)}$$

Nonadiabatic dynamics at TDDFT level

Why ?

The only pratical method for large systems.

- Efficiency and *accuracy* (?)
- Available package
- Map to the CIS form easily
- Potential Possiblilty on GPU computation

Challenging ?

- Nonadiabatic couplings

  Numerical v.s. analytical ones

•Charge-transfer problems

•Single-reference problems: S0/S1 Crossing

•Other problems: double excitations...

TD-DFT

- how to get non-adiabatic coupling vector
- numerical

$$F_{ij}\left(t+\frac{\Delta t}{2}\right)=\frac{\int d\mathbf{r}\phi_j^*(t)\phi_i(t+\Delta t)-\int d\mathbf{r}\phi_j^*(t+\Delta t)\phi_i(t)}{2\Delta t}.$$

$$\left|\Psi_i\right\rangle=\sum_k c_k{}^i\left|\Phi_i{}^{CSF}\right\rangle \qquad \begin{pmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B} & \mathbf{A}\end{pmatrix}\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix}=\omega\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}\mathbf{X}\\ \mathbf{Y}\end{pmatrix}$$

J. Pittner et al. Chem. Phys. 356 (2009) 147–152
I. Tavernelli et al. J. Mol. Struct. Theochem 914 (2009) 22-29

Numerical nonadiabatic couplings

AO Overlap: $< \mu_p^{AO}(t) | \mu_q^{AO}(t+\delta t) >$

MO Overlap: $< \kappa_m^{MO}(t) | \kappa_n^{MO}(t+\delta t) >$

CSF Overlap (Overlap of Slater determinant)

$$< \psi_k^{CSF}(t) | \psi_l^{CSF}(t+\delta t) >$$

Overlap of Electronic Wavefunctions

$$< \phi_j(t) | \phi_i(t+\delta t) >$$

Nonadiabatic couplings

$$F_{ji} = \frac{1}{\delta t} < \phi_j(t) | \phi_i(t+\delta t) >$$

Algorithms

1. Initialization of velocity, gradients, and quantum amplitudes
2. Time propagation of coordinates and velocities on the selected PES.
3. Computation of energies, gradients, and nonadiabatic coupling vectors of all relevant states at new position and velocity.
4. Time propagation of quantum amplitudes and computation of hopping probabilities.
5. Random number generation and comparison with hopping probabilities
6. If hopping is rejected: inversion of velocity .
7. If hopping is performed: velocity adjustment and update of the active PES for molecular dynamics.
8. Back to point 2.

Introduction to the codes

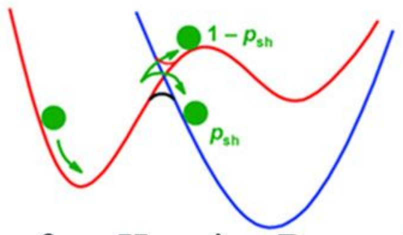
Excited State Method

CASSCF, ADC(2), TDDFT, CIS, TDDFTB

$$\hat{H}_e\, \chi(\mathbf{r},\mathbf{R}) = E_e(\mathbf{R})\, \chi(\mathbf{r},\mathbf{R})$$

$$\mathbf{HC} = \mathbf{SCE}$$

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \omega \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix}$$

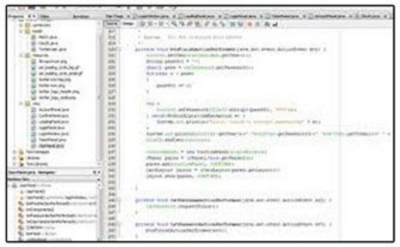


Surface Hopping Dynamics

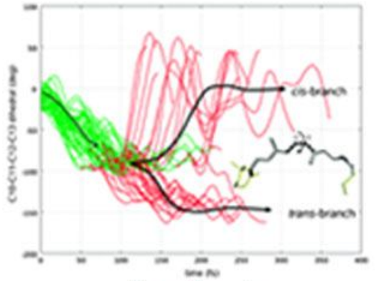
$$i\hbar \frac{dc_j(t)}{dt} = \sum_i c_i(t)[H_{ji} - i\hbar \dot{\mathbf{R}} \cdot \mathbf{d}_{ji}]$$

$$\mathbf{d}_{ji} \equiv \int d\mathbf{r} \phi_j^*(\mathbf{r},\mathbf{R})[\nabla_{\mathbf{R}} \phi_i(\mathbf{r},\mathbf{R})]$$

$$P_{ij} = \frac{2\int_t^{t+\Delta t} dt \mathrm{Re}(c_i^* c_j \dot{\mathbf{R}} \cdot \mathbf{d}_{ji})}{|c_i(t)|^2}$$



Code development



Analysis

code management

- version management：git
- code management：Makefile

Code development and feature

- **On-the-fly dynamics surface hopping dynamics**
- the velocity-Verlet algorithm;
- fewest switches algorithms
- Langevin thermostat
- **Interfaces**
- TURBOMOLE (CIS, TD-DFT, RI-CC2, ADC(2))
- GAUSSIAN (CIS, TD-DFT)
- GAMESS (CIS TD-DFT)
- Q-CHEM (in progress)
- Molpro (in progress)
- **Initial conditions**
- Wigner distribution
- **enviroment effects**
- Langevin dynamics
- effective fragment potential (interface with Gamess-US & Q-CHEM) in process
- **Output and analysis**
- Statistical analysis of results
- transition density analysis tools

how-to: install

- See INSTALL file in doc/ directory

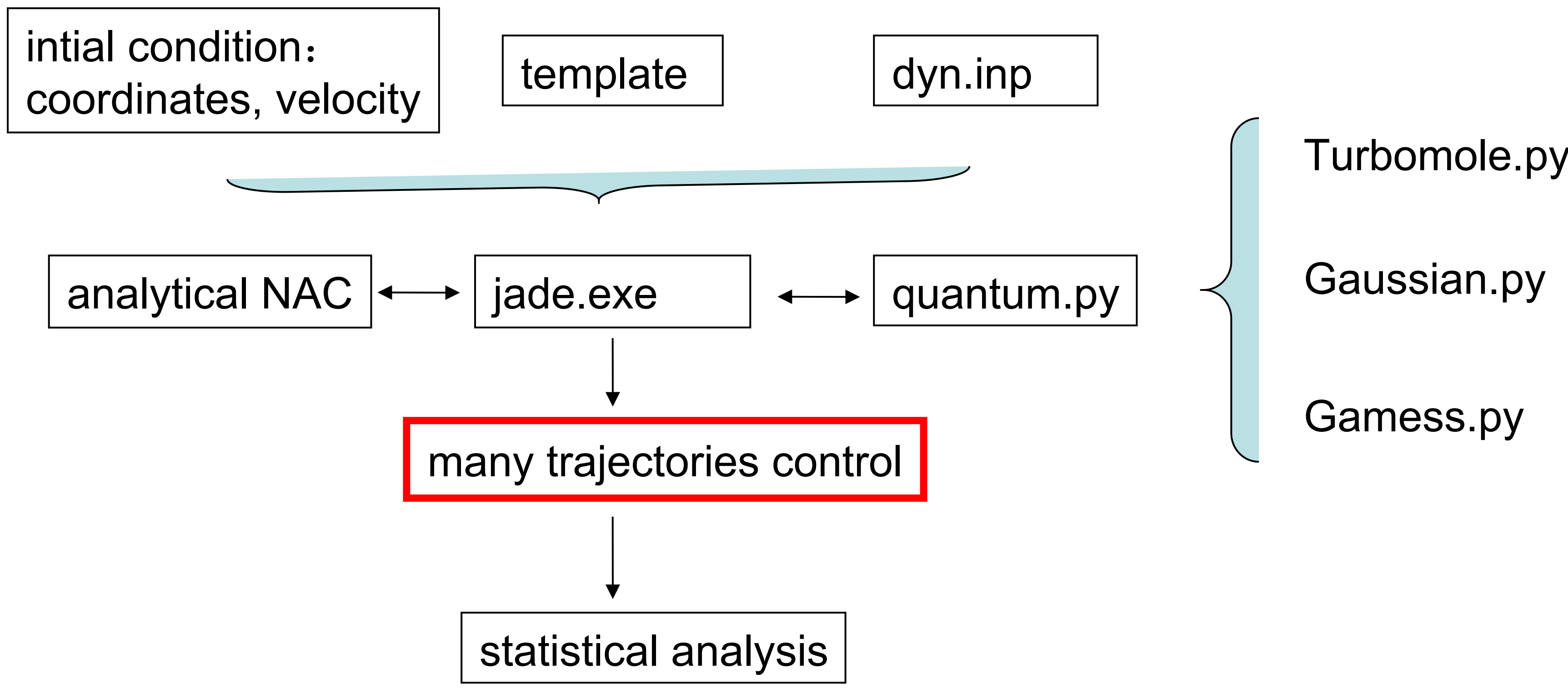
- To compile the Fortran90 programs of the JADE package, go to the src/ directory.

- go to *src* directory and run
  make; make install;

- if the compilation is successful
- then, set the environment variable $JADE_HOME

- export **JADE_HOME**=/home/jade-package
- source $JADE_HOME/bin/JADERC

code


intial condition：
coordinates, velocity
template
dyn.inp
analytical NAC
jade.exe
quantum.py
Turbomole.py
Gaussian.py
Gamess.py
many trajectories control
statistical analysis

how-to: initial conditions

wigner sampling

$SH_HOME/wigner

1. frequency calcuation: gaussian/turbomole/mndo

prepare.py

2. sampling process

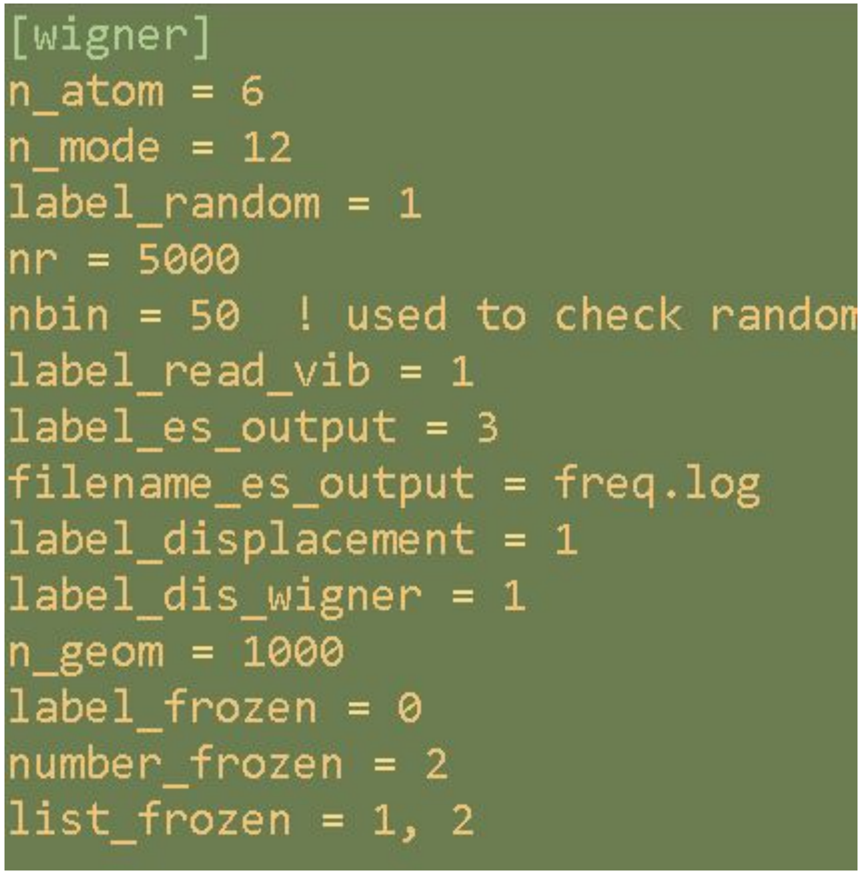
sampling.py

3. filter a list of sample geometries

mdfilter.py

The transition probability is calculated as P = (f/DE^2) / max(f/DE^2)

how-to: wigner sampling



```
[wigner]
n_atom = 6
n_mode = 12
label_random = 1
nr = 5000
nbin = 50  ! used to check randon
label_read_vib = 1
label_es_output = 3
filename_es_output = freq.log
label_displacement = 1
label_dis_wigner = 1
n_geom = 1000
label_frozen = 0
number_frozen = 2
list_frozen = 1, 2
```

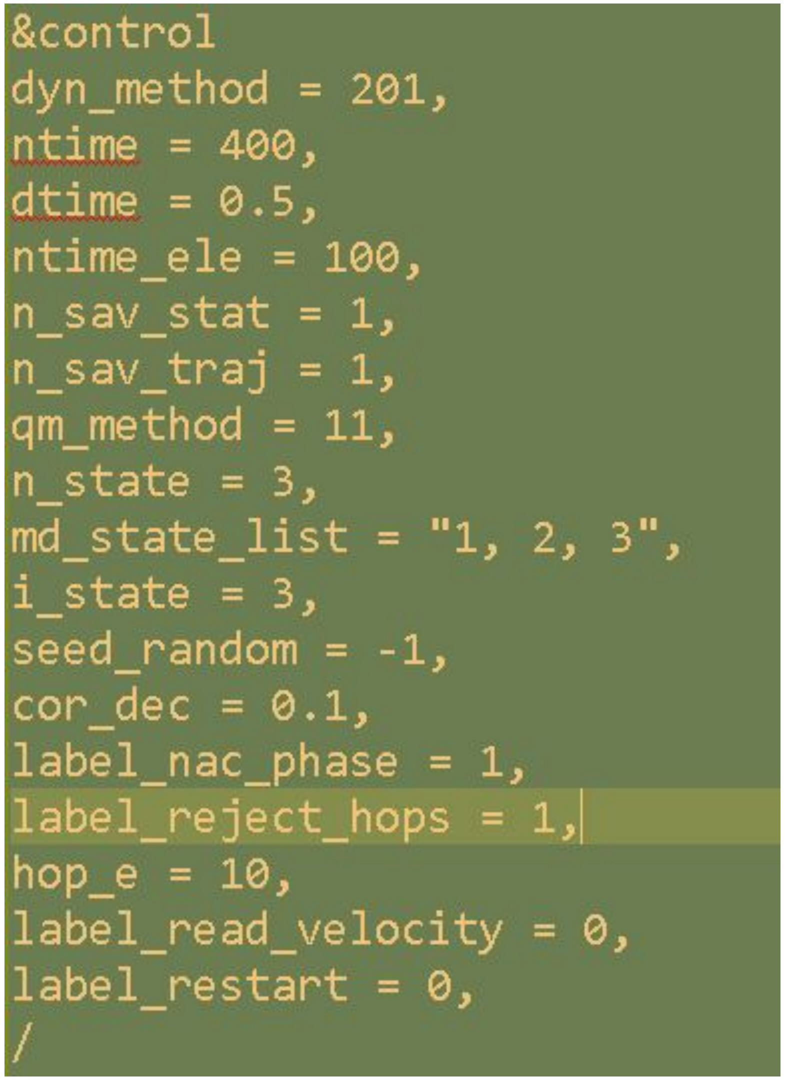
how-to: template

- td-dft
- support: gaussian/turbomole/gamess
- $SH_HOME/test

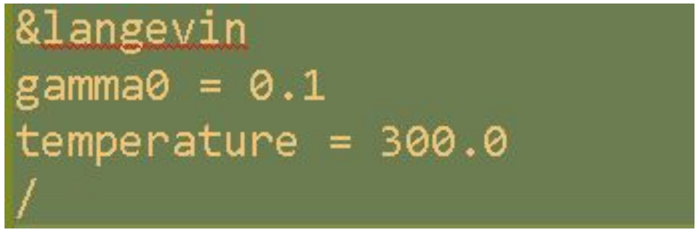
how-to: dyn.inp

- **FORTRAN-NAMELIST-STYLE**
- up to now, include three sections:
- &control
- &quantum
- &langevin
- ...

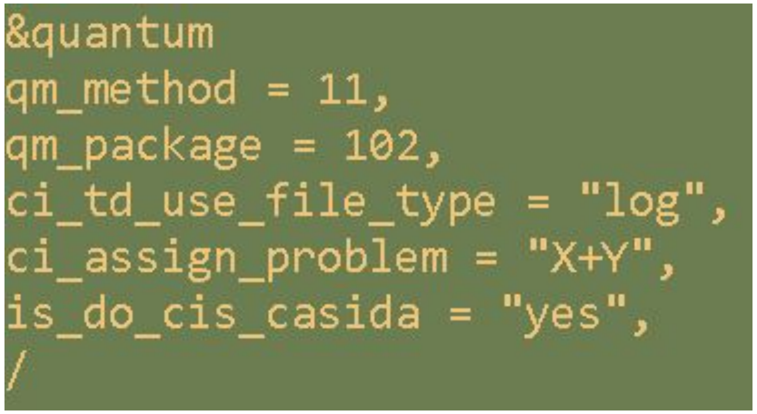
how-to: dyn.inp



```
&control
dyn_method = 201,
ntime = 400,
dtime = 0.5,
ntime_ele = 100,
n_sav_stat = 1,
n_sav_traj = 1,
qm_method = 11,
n_state = 3,
md_state_list = "1, 2, 3",
i_state = 3,
seed_random = -1,
cor_dec = 0.1,
label_nac_phase = 1,
label_reject_hops = 1,
hop_e = 10,
label_read_velocity = 0,
label_restart = 0,
/
```



```
&langevin
gamma0 = 0.1
temperature = 300.0
/
```



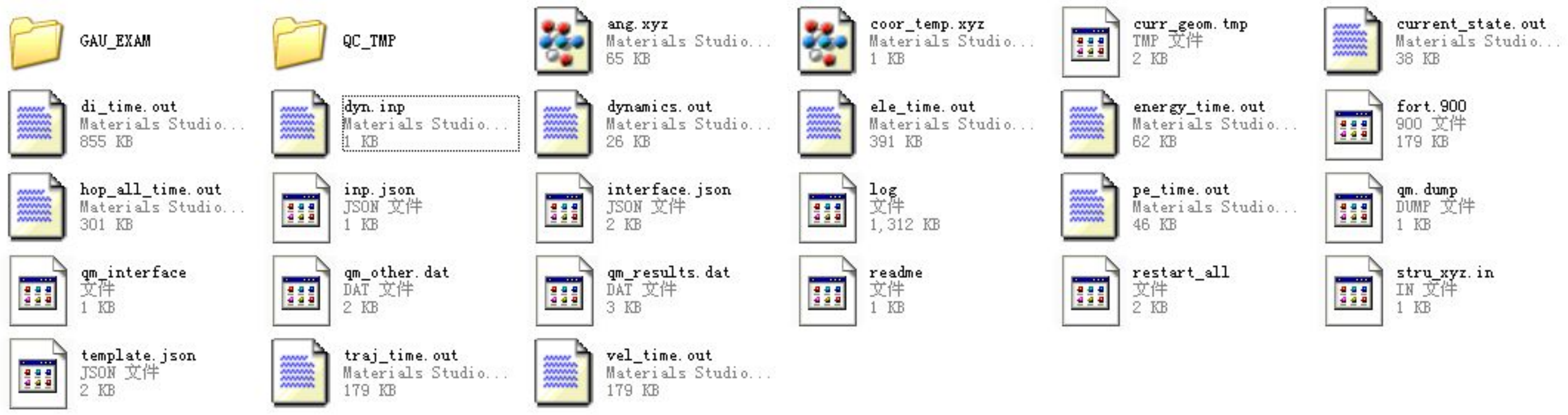
```
&quantum
qm_method = 11,
qm_package = 102,
ci_td_use_file_type = "log",
ci_assign_problem = "X+Y",
is_do_cis_casida = "yes",
/
```

restartable

- restartable
- set label_restart to 1

directory in dynamics

jade.exe > abc.log &



```
GAU_EXAM            QC_TMP              ang.xyz              coor_temp.xyz        curr_geom.tmp       current_state.out
                                        Materials Studio...  Materials Studio...  TMP 文件            Materials Studio...
                                        65 KB                1 KB                 2 KB                38 KB

di_time.out         dyn.inp             dynamics.out         ele_time.out         energy_time.out     fort.900
Materials Studio... Materials Studio... Materials Studio...  Materials Studio...  Materials Studio... 900 文件
855 KB              1 KB                26 KB                391 KB               62 KB               179 KB

hop_all_time.out    inp.json            interface.json       log                  pe_time.out         qm.dump
Materials Studio... JSON 文件           JSON 文件            文件                 Materials Studio... DUMP 文件
301 KB              1 KB                2 KB                 1,312 KB             46 KB               1 KB

qm_interface        qm_other.dat        qm_results.dat       readme               restart_all         stru_xyz.in
文件                DAT 文件            DAT 文件             文件                 文件                IN 文件
1 KB                2 KB                3 KB                 1 KB                 2 KB                1 KB

template.json       traj_time.out       vel_time.out
JSON 文件           Materials Studio... Materials Studio...
2 KB                179 KB              179 KB
```

how-to: many-trajectory control

- write your own scripts....
- manyjob.py

how-to: statistical analysis

- see $SH_HOME/contrib
- jobstatus.py
- jobfilter.py

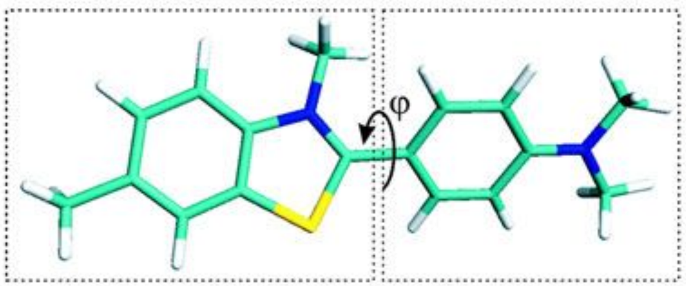
- structural analysis：geom_time.py
- energy analysis：pe_time.py
- population analysis：state_population.py
- spectrum analysis ...

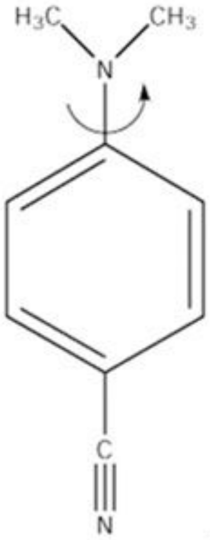
examples

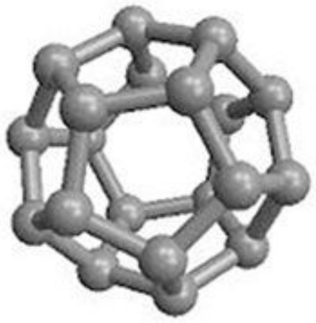
- simple examples
- $SH_HOME/test

(a)

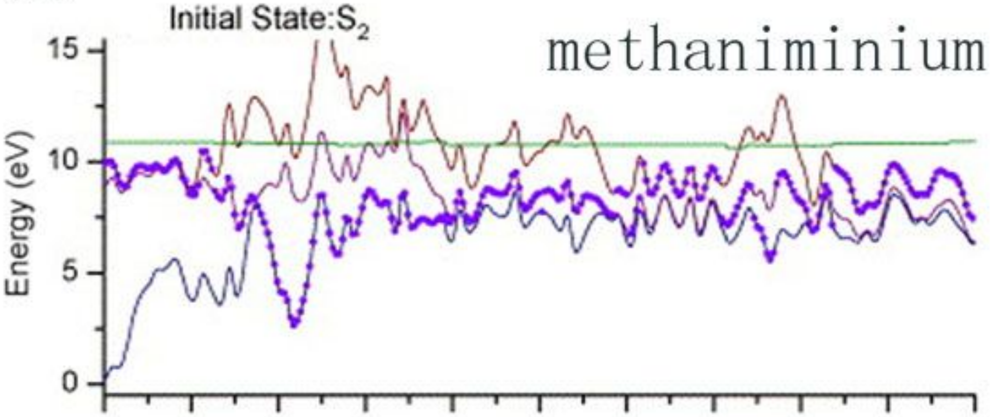
methaniminium

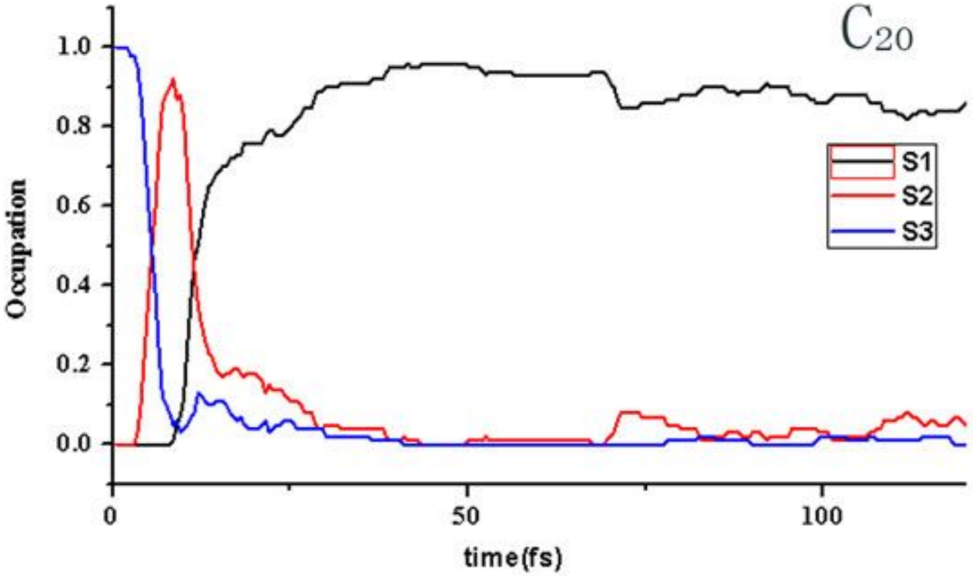




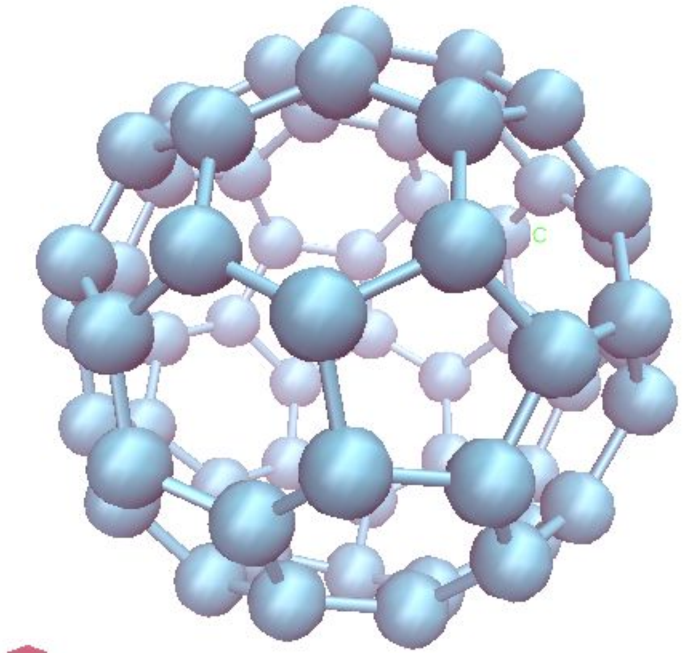


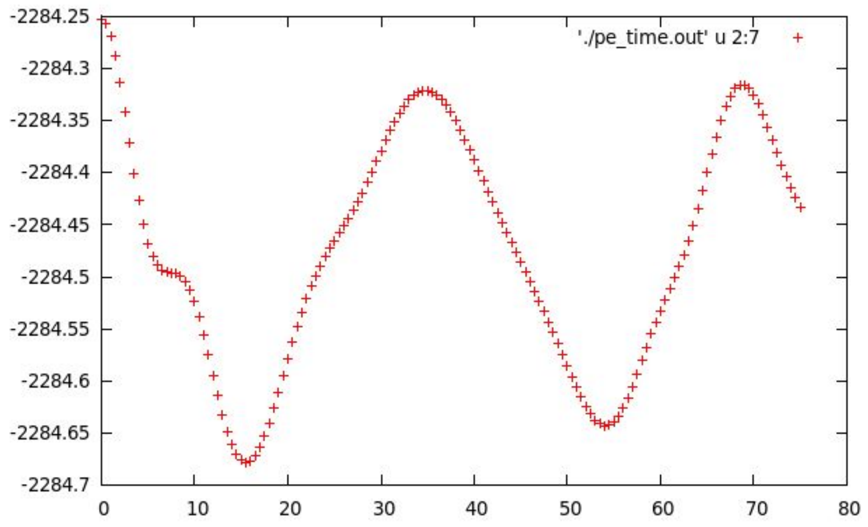
(b)

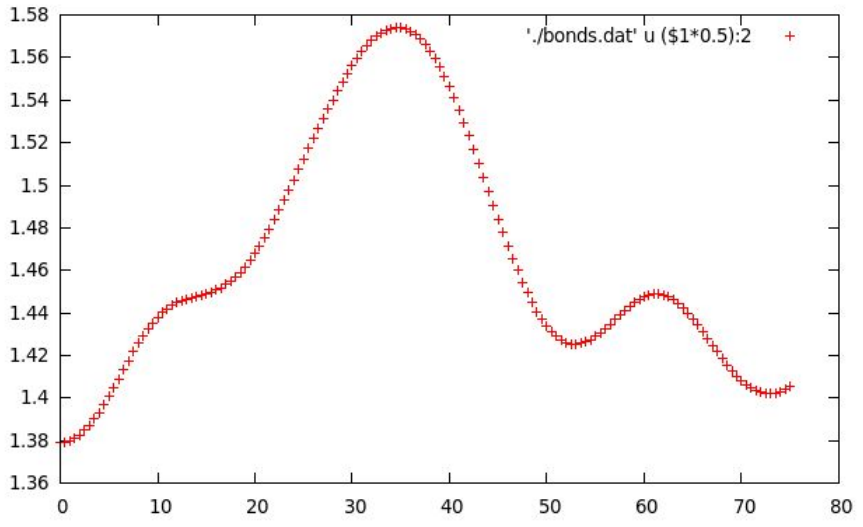




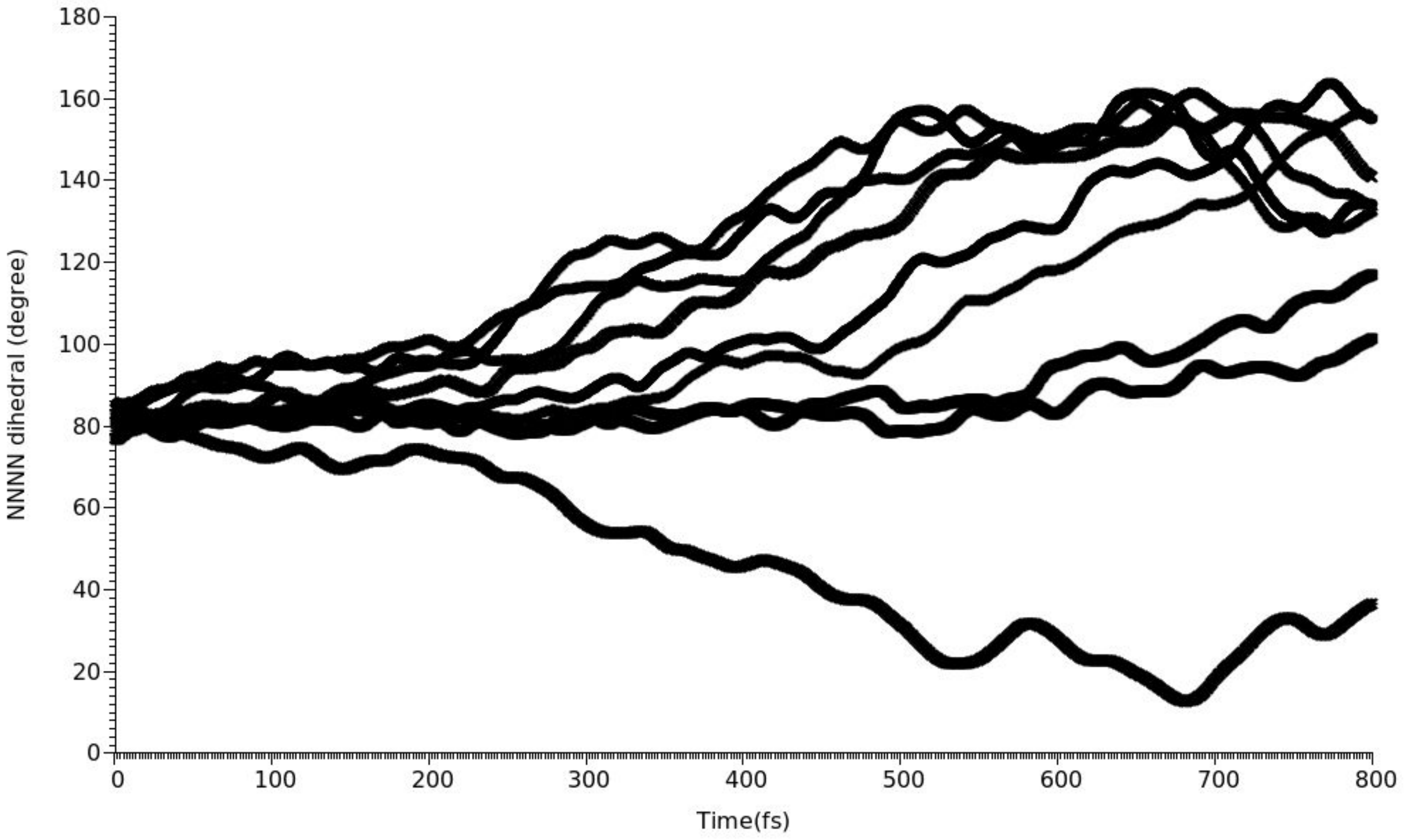
C60







turbomole; tddft/pbe;
n_state=6; i_state=6



180
160
140
120
100
80
60
40
20
0
NNNN dihedral (degree)
0
100
200
300
400
500
600
700
800
Time(fs)

on-going

- analysis tools
- Q-CHEM interface..
- MCSCF with analytical NAC

- DFTB
- QM/MM
- open shell

Thanks