Time Dependent *ab initio* Package (TDAP) —— Introduction And Practice

Speaker：Peiwei You

Supervisor：Sheng Meng

July. 24, 2019

中国科学院物理研究所
Institute of Physics Chinese Academy of Sciences

CONTENTS

1.1.1 Introduction to SIESTA

SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms) is both a method and its computer program implementation, to perform electronic structure calculations and *ab initio* molecular dynamics simulations of molecules and solids. (https://departments.icmab.es/leem/siesta/)

Its main characteristics are:

- **Geometry relaxation, fixed or variable cell.**
- **Constant-temperature molecular dynamics.**
- **Variable cell dynamics.**
- **Spin polarized calculations.**
- **k-sampling of the Brillouin zone.**
- **Local and orbital-projected density of states.**
- **Dielectric polarization.**
- **Vibrations (phonons).**
- **Band structure.**
- **Total and partial energies.**
- **Atomic forces.**
- **Stress tensor.**
- **Electric dipole moment.**
- **Atomic, orbital and bond populations.**
- **Electronic density.**

1.1.2 Compilation of SIESTA

- **Copy files:**
  sh ../Src/obj_setup.sh

- **Configuration and generate arch.make:**
  ../Src/configure --enable-mpi

- **Modify arch.make**
  choose a compiler: FC=mpif90
  static library link：
  BLAS_LIBS=/home/pwyou/mathlib/lib/librefblas.a
  LAPACK_LIBS=/home/pwyou/mathlib/lib/libreflapack.a
  BLACS_LIBS=
  SCALAPACK_LIBS=/home/pwyou/mathlib/lib/libscalapack.a
- **Compile and link**
  make

1.1.3 Input File of SIESTA

- **An input file, called input.fdf**

  Written in Flexible Data Format

  Contains: Physical data of the system

  Variables to control the approximations

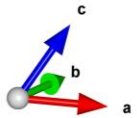
- **A pseudopotential file for all different elements in the input file**

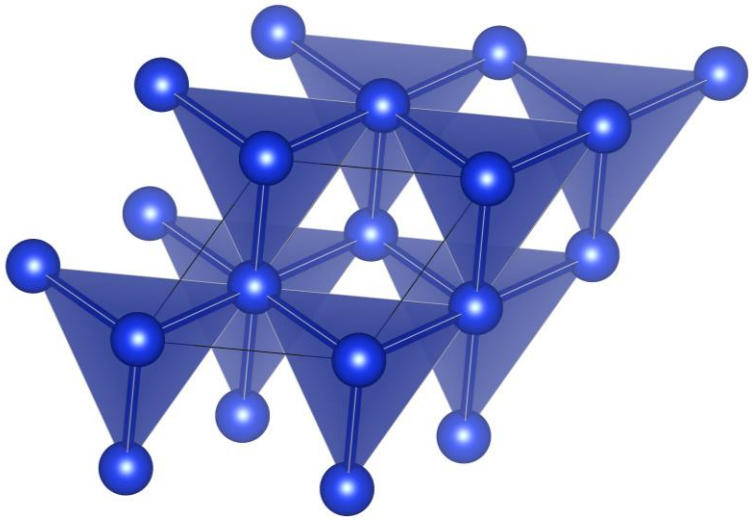
  Formatted ASCII (.psf): H.psf, O.psf

Put input file in one directory.

1.2.1 Atomic Structure

```
AtomicCoordinatesFormat   Fractional
LatticeConstant  1.0  Ang
NumberOfAtoms 2
NumberOfSpecies 1
%block ChemicalSpeciesLabel
 1  14  Si
%endblock ChemicalSpeciesLabel
%block LatticeVectors
2.715000 2.715000 0.000000
0.000000 2.715000 2.715000
2.715000 0.000000 2.715000
%endblock LatticeVectors
%block AtomicCoordinatesAndAtomicSpecies
   0.00  0.00  0.00   1  Si       1
   0.25  0.25  0.25   1  Si       2
%endblock AtomicCoordinatesAndAtomicSpecies
```





1.2.2 K Mesh

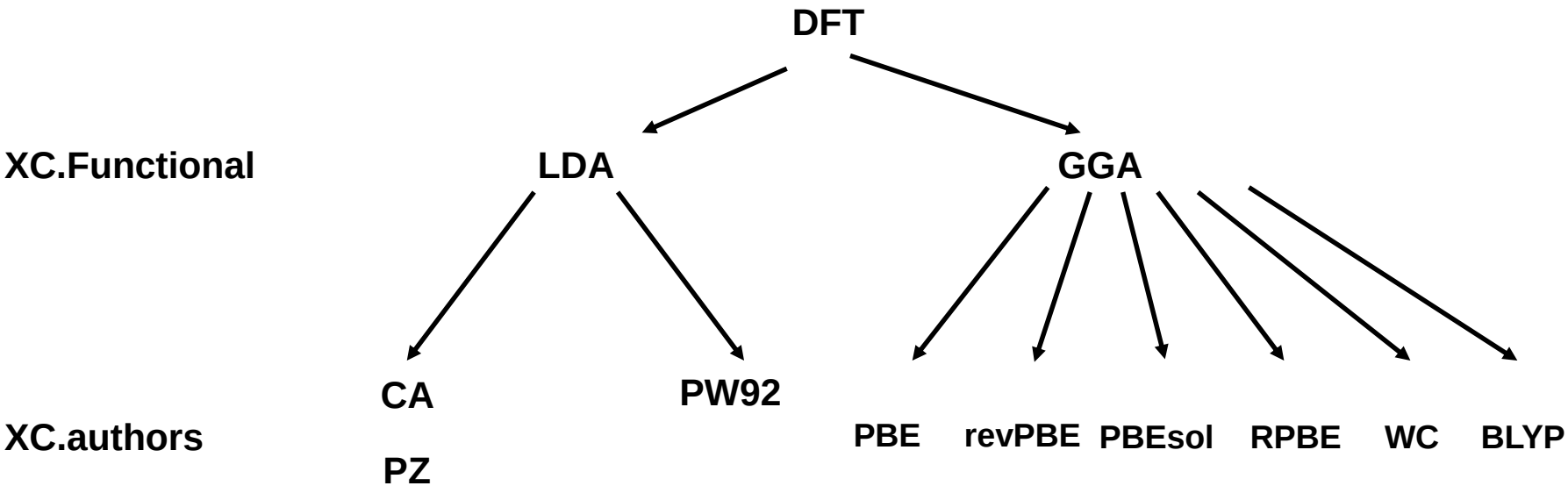
- **Monkhorst_Pack sampling**

```
%block kgrid_Monkhorst_Pack
7 0 0 0.0
0 7 0 0.0
0 0 7 0.0
%endblock kgrid_Monkhorst_Pack
```

- **Band lines model**

```
BandLinesScale  pi/a
%block BandLines
 1  0.000  0.000  0.000     \Gamma   # Begin at Gamma
25  2.000  0.000  0.000     X     # 25 points from Gamma to X
10  2.000  1.000  0.000     W     # 10 points from X to W
15  1.000  1.000  1.000     L     # 15 points from W to L
20  0.000  0.000  0.000     \Gamma   # 20 points from L to Gamma
25  1.500  1.500  1.500     K     # 25 points from Gamma to K
%endblock BandLines
```

1.2.3 Exchange And Correlation Functional



CA ≡ **Ceperley-Alder**

PZ ≡ **Perdew-Zunger**

PW92 ≡ **Perdew-Wang-92**

PBE ≡ **Perdew-Burke-Ernzerhof**

WC ≡ **Wu-Cohen**

BLYP ≡ **Becke-Lee-Yang-Parr**

1.2.4 Convergence Setup

- **Parameter**

SCFMustConverge T
DM.MixingWeight 0.3
DM.Tolerance 1.d-4
SCF.Mixer.Method Pulay|Broyden|Linear
DM.NumberPulay 5
SCF.Pulay.Damping 0.5

- **Linear**

$$\boldsymbol{\rho}_{\mathrm{in}}^{n+1} = \boldsymbol{\rho}_{\mathrm{in}}^{n} + w\,\mathrm{R}[n].$$

- **Pulay**

$$\boldsymbol{\rho}_{\mathrm{in}}^{n+1} = \boldsymbol{\rho}_{\mathrm{in}}^{n} + G\,\mathrm{R}[n] + \sum_{i=n-N+1}^{N-1} \alpha_i(\mathrm{R}[i] + G\,\Delta\mathrm{R}[i])$$

1.2.5 Molecular Dynamics

- **Two major types of MD**

MD.TypeOfRun	**cg**	**MD.TypeOfRun**	**verlet**
MD.NumCGsteps	**800**	**MD.FinalTimeStep**	**1**
MD.MaxCGDispl	**0.1 Ang**	**MD.LengthTimeStep**	**1.0 fs**
MD.MaxForceTol	**0.01 eV/Ang**	**MD.InitialTemperature**	**0 k**

- **Other MD.TypeOfRun**

Broyden
FIRE
Nose
ParrinelloRahman
FC

1.2.6 Run SIESTA

```
#!/bin/bash
#SBATCH -J siesta
#SBATCH -N 1
#SBATCH --ntasks-per-node=36
#SBATCH -p regular
#SBATCH -t 00:20:00
pwd | cat >> pwd.dat

module load mpi/mvapich2/gnu/2.3b
module load apps/siesta/gnu/4.1-b4
EXEC=siesta
srun --mpi=pmi2 $EXEC < input.fdf > result
```

sbatch run-siesta.sh

1.3.1 Example- K Mesh -Silicon

%block kgrid_Monkhorst_Pack #only gamma k point
x 0 0 0.0
0 x 0 0.0
0 0 x 0.0
%endblock kgrid_Monkhorst_Pack

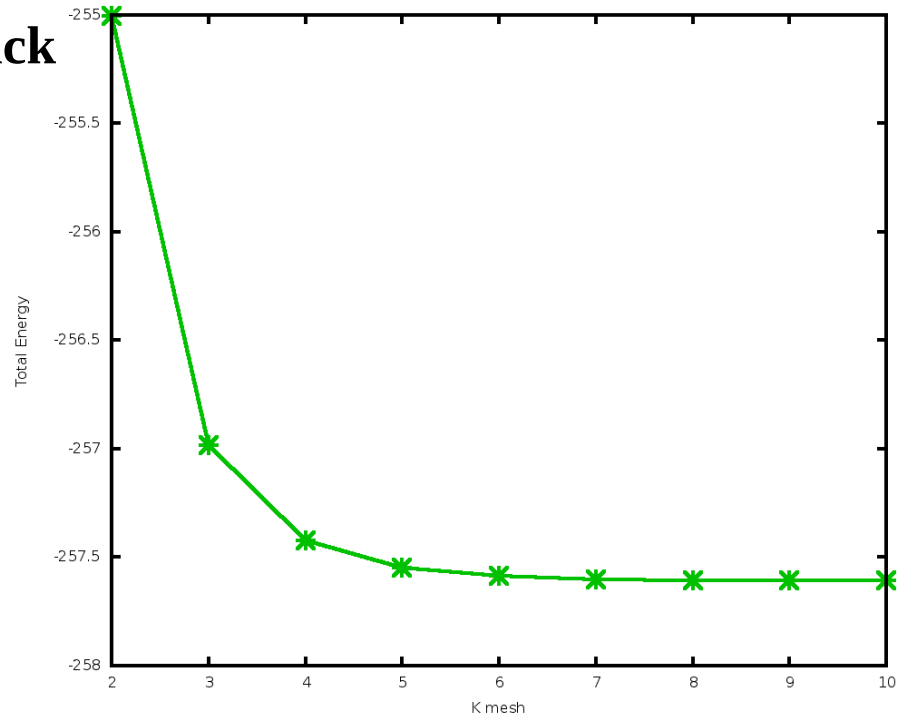
Command:

sh test-K.sh

./get-energy.py

gnuplot plot-energy.gnu

download ek.png



1.3.2 Example - Energy Cutoff

MeshCutoff x Ry

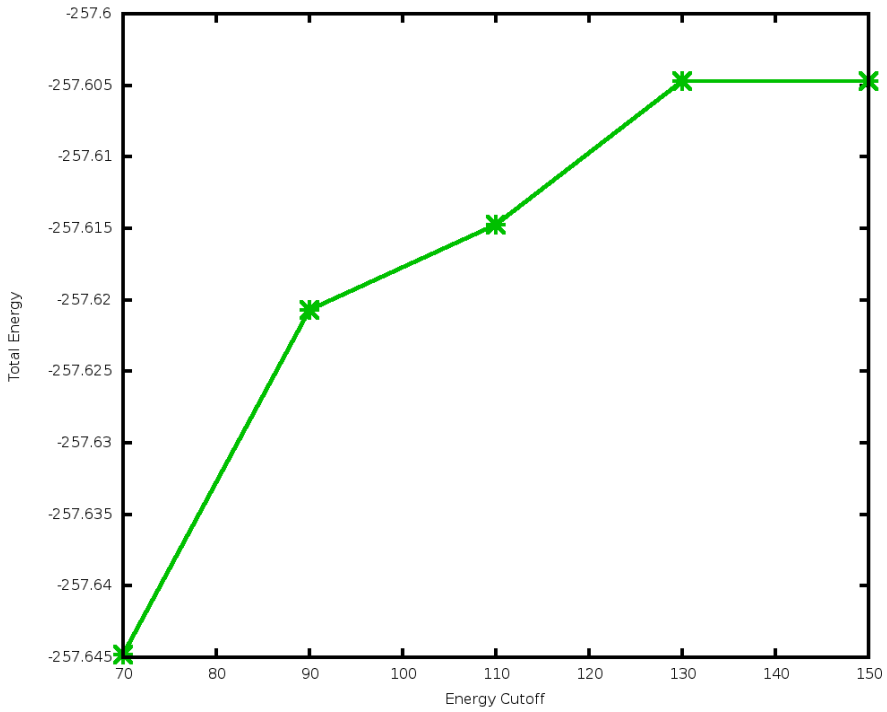
Command:

sh test-Ecut.sh

./get-energy.py

gnuplot plot-energy.gnu

download ek.png

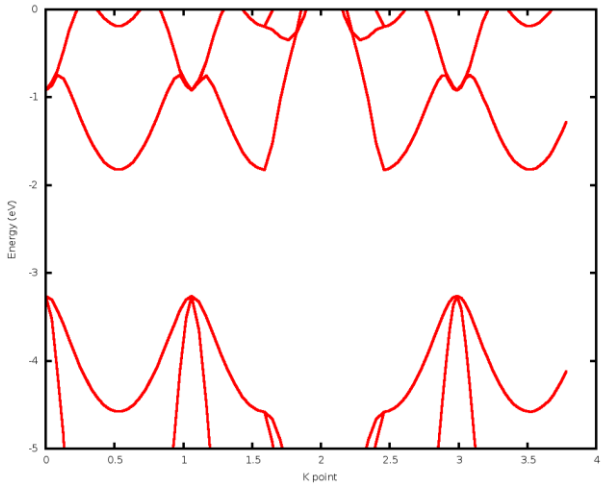


1.3.3 Example - Band Structure

```
BandLinesScale  ReciprocalLatticeVectors
%block BandLines          # These are comments
 1   0.000  0.000  0.000    \Gamma   # Begin at Gamma
50   1.000  0.000  0.000    X     # 50 points from Gamma to X
50   1.000  0.500  0.000    W     # 50 points from X to W
50   0.500  0.500  0.500    L     # 50 points from W to L
50   0.000  0.000  0.000    \Gamma   # 50 points from L to Gamma
50   0.750  0.750  0.750    K     # 50 points from Gamma to K
%endblock BandLines
```

Command:

sbatch run-siesta.sh
./gnubands siesta.bands > bandstructure
gnuplot plot-bandstructure.gnu
download band.png



1.3.4 Example - MD Trajectory, Energy, Temperature, Pressure

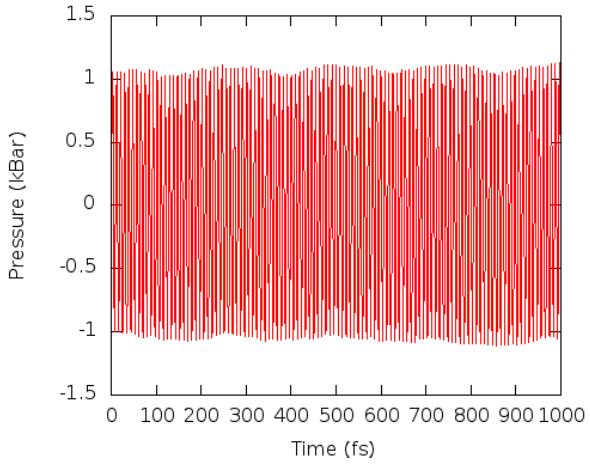
Command:

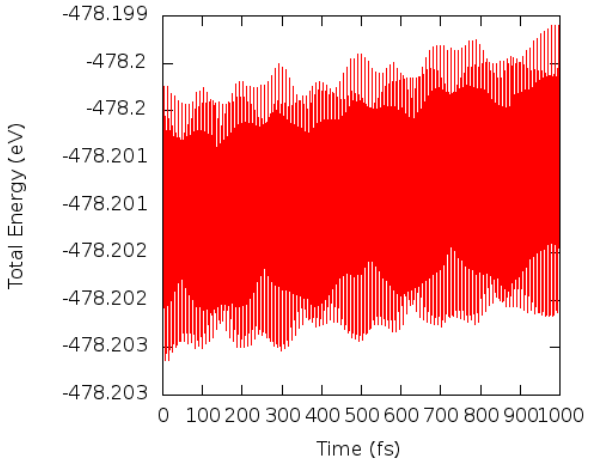
sbatch run-siesta.sh

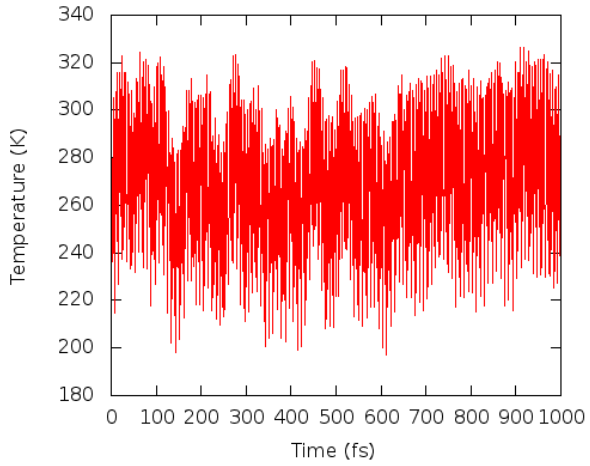
./get-bondlength.py

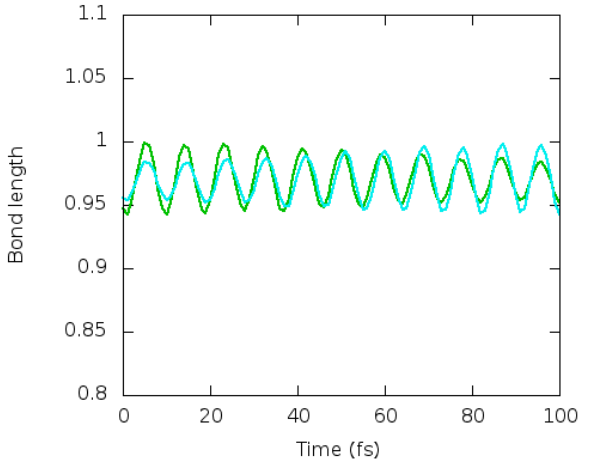
gnuplot plot-MD-E-T-P.gnu

download MD-E-T-P.png









1.3.5 Save Data For Post-processing

Information needed as input for various post-processing programs,

for example, to visualize:

	FDF tag to save file	Name of output file
Total charge density:	**SaveRho**	**SystemLabel.RHO**
Deformation charge density:	**SaveDeltaRho**	**SystemLabel.DRHO**
Electrostatic potential:	**SaveElectrostaticPotential**	**SystemLabel.VH**
Total potential:	**SaveTotalPotential**	**SystemLabel.VT**
Local density of states:	**LocalDensityOfStates**	**SystemLabel.LDOS**
Charge density contours:	**WriteDenchar**	**SystemLabel.DIM**
Atomic coordinates:	**WriteCoorXmol**	**SystemLabel.xyz**
Animation of a molecular dyn:	**WriteMDXMol**	**SystemLabel.ANI**

2.0 TDAP

http://tdap.iphy.ac.cn/

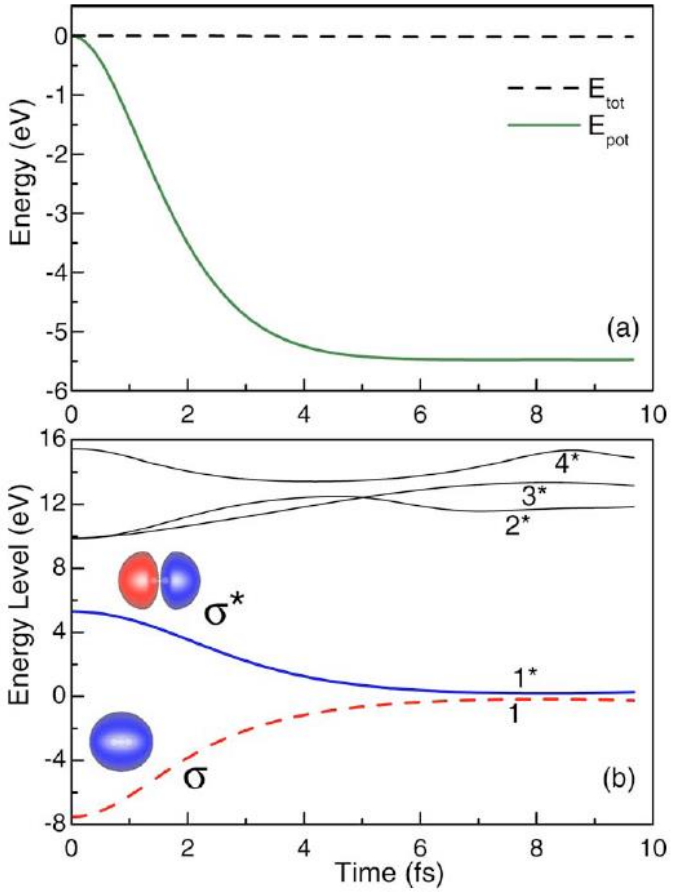
2.1 Time Evolved Equation

$$i\hbar\frac{\partial c}{\partial t} = S^{-1}Hc$$

$$\phi_j = \sum_i c_i^j \mu_i , S_{ij} = \langle \mu_i | \mu_j \rangle$$

$$c(t+\Delta t) = U(t+\Delta t, t)c(t)$$
$$= \hat{T}exp\left(-i\hbar \int_t^{t+\Delta t} S^{-1}(t')H(t')dt'\right)c(t)$$

$$c(t+\Delta t) = \frac{1 - i\hbar S^{-1}(t)H(t)\Delta t/2}{1 + i\hbar S^{-1}(t)H(t)\Delta t/2} \cdot c(t)$$



Meng, S.; Kaxiras, E. *The Journal of chemical physics* **2008,** *129* (5), 054110.

2.2.1 Solution and K Points

SolutionMethod (String): Character string to choose between diagonalization (diagon) or Order-N (OrderN) solution of the Hamiltonian or the TDDFT solver (evolve).

Note: Always use evolve to do TDDFT calculations, otherwise all the parameters listed below will not be read.

Default value: diagon

Kpoints: only Gamma point

```
%block kgrid_Monkhorst_Pack
1 0 0 0.0
0 1 0 0.0
0 0 1 0.0
%endblock kgrid_Monkhorst_Pack
```

2.2.2 Population Switch

TD.PopulationTransition (data block): Includes arbitrary number of lines. Each line with the format :
StartBand – EndBand - JumpedElectrons
StartBand is the band from which the electron is excited. If 0, no excitation occurs.
EndBand is the band to which the electron is excited to.
JumpedElectrons is the number of electrons pumped from StartBand to EndBand.
For instance:

```
%block TD.PopulationTransition
-1 1 0.5
-1 2 1.0
%endblock TD.PopulationTransition
```

means that 0.5 and 1.0 electrons are pumped from the HOMO band to the LUMO band, and from the HOMO to the LUMO+1 band, respectively.

2.2.3 Time Dependent Electric Field

$$E(t) = E_0 \cos(2\pi f t) \exp(-\frac{(t-t_0)^2}{2\sigma^2})$$

TD.DeltaElectricField (Logical): Specify a delta electric field. Then gaussian electric field will be ignored. Use for calculating absorption spectrum.
Default value: .false.

TD.LightFrequency (Real): The frequency f of the electric field.
Default value: 0.5 fs

TD.LightInitialTime (Real): The initial time t_0 to introduce the electric field.
Default value: 100.0 fs

TD.LightTimeScale (Real): The peak width of the wave package.
Default value: 25.0 fs

2.2.4 Band Projection

TD.WriteDMOfSelectiveOrbitals (logical): Whether to output the partial DM projected to the orbitals/bands specified.
Default value: false
TD.PartialDMBand (data block): Specify the bands to do the projection. See also TD.PartialDMSumOrbitals.
Multiple lines with each of them includes one band index. The corresponding charge density will be projected to the selected orbital blocks, and output to the file: chgBand-Index.txt
For instance:

```
%block TD.PartialDMBand
1
2
%endblock TD.PartialDMBand
```

will analyze the band No. 1 and No. 2, and project them to the orbital blocks.

2.2.4 Band Projection

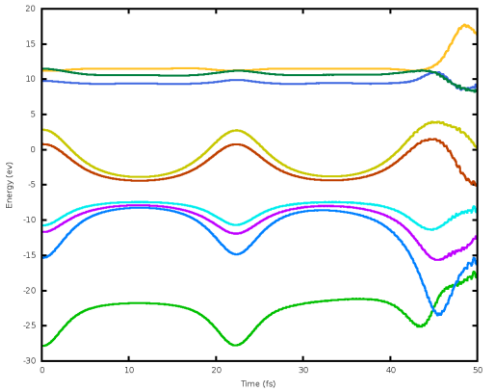
TD.PartialDMSumOrbitals (data block): Specify the orbital blocks for projection. See also TD.PartialDMBand.
Multiple lines with each of them include the start index and the end index of the orbital. The corresponding charge density will be projected to the selected orbital blocks, and output to file: chgBand-Index.txt
For instance:

```
%block TD.PartialDMSumOrbitals
1 10
11 22
%endblock TD.PartialDMSumOrbitals
```

will analyze the selected bands (using TD.PartialDMBand) and projected them onto the two orbital blocks: orbitals No. 1 to No. 10, and No. 11 to No. 22.

2.3.1 Example - Population Switch

```
TD.LengthTimeStep   0.01 fs
TD.FinalTimeStep    100
TD.WriteDMOfSelectiveOrbitals True
%block TD.PartialDMBand
1
4
5
%endblock TD.PartialDMBand
%block TD.PopulationTransition
-1 2 0.5
-1 1 0.5
%endblock TD.PopulationTransition
%block TD.PartialDMSumOrbitals
1 4
5 10
%endblock TD.PartialDMSumOrbitals
```



Command:

sbatch run-tdap.sh

./get-band.py

gnuplot plot-t-band.gnu

download t-band.png

2.3.2 Example – Time Dependent Electric Field

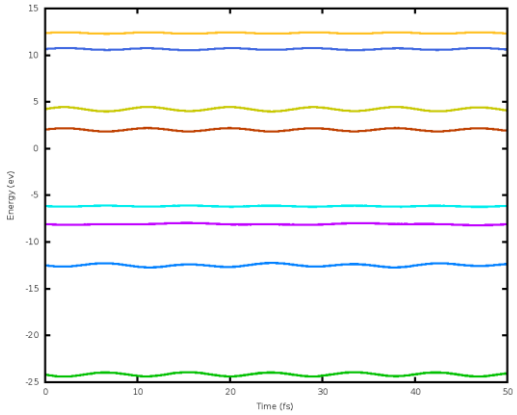
TD.LengthTimeStep 0.01 fs
TD.FinalTimeStep 100

%block ExternalElectricField
0.01 0.0 0.0 Ry/Bohr/e
%endblock ExternalElectricField

TD.LightFrequency 1.932 fs

TD.LightInitialTime 10.66 fs

TD.LightTimeScale 6 fs



Command:

sbatch run-tdap.sh

./get-band.py

gnuplot plot-t-band.gnu

download t-band.png

Thanks For Listening