

# DynEMol code

# Dynamics of Electrons in Molecules



Luis G. C. Rego

Department of Physics

Federal University of Santa Catarina

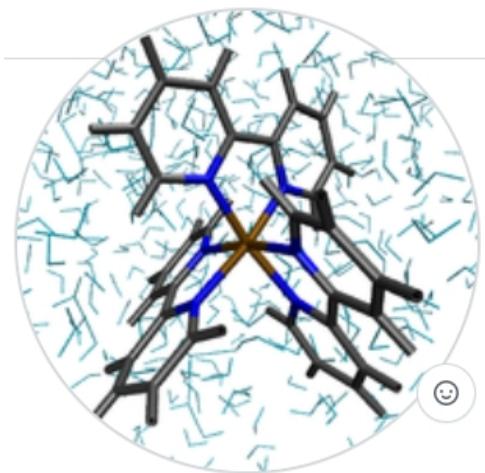
# Dynemol Tutorial

## Contents:

• General Information.....	3
• File card.inpt for execution directions.....	8
• Running Dynemol.....	15
• Examples:	
DNA in solution.....	18
NA-Dynamics in Solution.....	22
Dye-sensitized semiconductor.....	28
Charge Transfer vs Structural Relaxation.....	41
6x6x6 MAPbI <sub>3</sub> hydrid perovskites .....	51

# Dynemol: general information

[github.com/lgrego/Dynemol](https://github.com/lgrego/Dynemol)



**DynEMol: Dynamics of Electrons in Molecules**  
lgrego

Department of Physics, Federal University...

luis.gc.rego@gmail.com

<http://luisrego.sites.ufsc.br/>

master 3 branches 0 tags

Switch branches/tags X

Find or create a branch...

Branches Tags

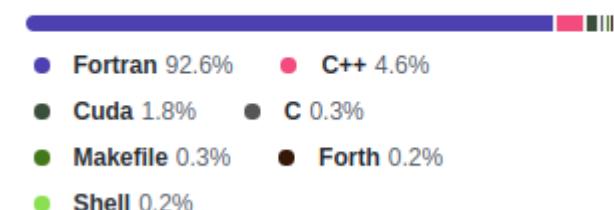
✓ master default

SingleNode

alpha-SO

[View all branches](#)

## Languages



# Dynemol: general information

## Supported branches:

- **Master**: hybrid MPI / openMP / GPU version
- **SingleNode**: openMP / GPU version
- **Alpha-SO**: alpha version of SingleNode with focus in Spin effects

# Compiling Dynemol

OS of choice: Linux

Compiler of choice:

Intel oneAPI 2022 = Base Toolkit + HPC Toolkit

Compilation options:

make (*dynemol*) - standard compilation

make *safe* - compilation with safe features

make *debug* - adds flag -g for debugging

make *serial* - remove all parallelization flags

make *gdb* - prepare code to GDB (equivalent to debug + serial) analysis

make *vtune* - prepare code to intel-Vtune analysis

make + *edit CUDA configurations in the makefile*

# Running Dynemol

Setup environment variables:

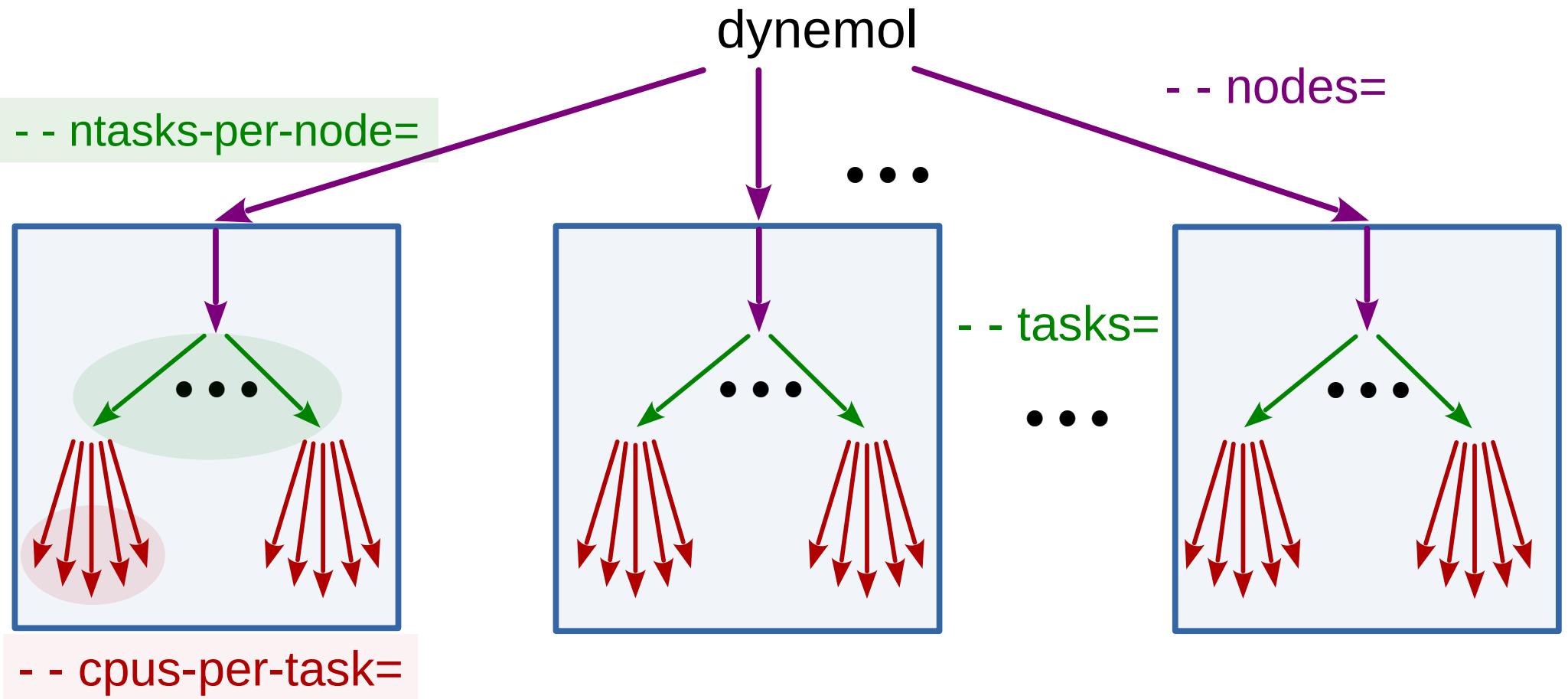
```
export DYNEMOLDIR= "path to dynemol executable directory"
```

```
export DYNEMOLWORKDIR= $(pwd)
```

In directory \$DYNEMOLWORKDIR:

- have the appropriate input files for the job
- for execution directions: edit file ***card.inpt***
- Run \$DYNEMOLDIR/dynemol

# Hybrid MPI + openMP mode on SLURM



```
> export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
```

```
> srun $DYNEMOLDIR/dynemol
```

# card.inpt

- Check the file “\$DYNEMOLDIR/card\_file\_formats” for guidance

For instance:

```
!-----  
! EXECUTION CONTROL  
  
DRIVER = ! <== q_dynamics , avrg_configs , Genetic_Algo , diagnostic , slice_{Cheb, A0, FSSH, CSDM} , MM_Dynamics  
  
Survival = ! <== .TRUE. for any dynamics simulation  
DP_Moment = ! <== .TRUE. or .FALSE. ; dipole moment fragment must be specified ad-hoc  
QMMM = ! <== .TRUE. for Non-Adiabatic simulations; couples electronic and nuclear dynamics  
OPT_parms = ! <== .TRUE. for reading OPT_basis parameters from "opt_eht_parms.input"  
ad_hoc = ! <== .TRUE. for using ad hoc tuning of parameters  
Band_structure = ! <== .TRUE. for static band-structure calculations  
  
!-----  
! STRUCTURE-FILE input FORMAT  
  
nuclear_matter = ! <== solvated_sys , extended_sys , MDynamics  
file_type = ! <== structure or trajectory ; default = structure  
file_format = ! <== xyz , pdb or vaspr ; default = pdb  
  
!-----  
! generate copies of the system by reflection  
  
nnx = 0 ; nny = 0 ! <== (nnx,nny) = (extended) REAL copies on each side  
! Integers, keep format ; default = (0,0)  
  
! Periodic Boundary Conditions  
  
PBC = [ 0 , 0 , 0 ] ! <== PBC replicas : 1 = yes , 0 = no  
! Integers, keep format , default = (0,0,0)
```

# card.inpt

- Check the file “\$DYNEMOLDIR/card\_file\_formats” for guidance

For instance:

```
|-----|  
| QDynamics parameters |  
|-----|  
t_i = ! <== default = 0.d0  
t_f = ! <== final time in PICoseconds (Real)  
n_t = ! <== number of time steps (Integer)  
  
n_part = ! <== # of particles to be propagated: default is e=1 , e+h=2 ; default = 2  
  
hole_state = char*3:Integer ! <== char*3 = 3-letter fragment or residue ; Integer = M0 of frag/res  
! <== GROUND STATE calcs = 0 (ZERO)  
! <== case STATIC & DP_calcs = hole state of special FMO  
! <== case DYNAMIC = intial M0 for < HOLE > wavepacket in DONOR fragment  
  
electron_state = char*3:Integer ! <== char*3 = 3-letter fragment or residue ; Integer = M0 of frag/res  
! <== case STATIC & DP_calcs = excited state of special FMO  
! <== case DYNAMIC = intial M0 for < ELECTRON > wavepacket in DONOR fragment  
  
LCMO = ! <== .TRUE. for initial wavepackets as Linear Combination of Molecular Orbitals (LCMO)  
! <== default = .FALSE.  
|-----|  
| SAMPLING parameters and SECURITY COPY |  
|-----|  
CT_dump_step = ! <== step for saving El&H1 survival charge density (Integer); default = 1  
frame_step = ! <== step for avrg_configs and time-slice dynamics ; frame_step <= size(trj) ; default = 1  
restart = ! <== .TRUE. for restarting dynamics  
step_security = ! <== step for saving backup files  
! <== default = 100 (QMMM) ; 1000 (MM)
```

# card.inpt

- Check the file “`$DYNEMOLDIR/card_file_formats`” for guidance

For instance:

```
!-----[REDACTED]-----!  
!-----[REDACTED]-----!  
!-----  
!----- DIAGNOSTIC & DATA-ANALYSIS & VISUALIZATION flags  
!  
HFP_Forces      = ! <== .TRUE. for QMMM calcs and .FALSE. otherwise; Hellman-Feynman-Pulay forces  
  
SPECTRUM        = ! <== .TRUE. for absorption spectrum calculations  
Alpha_Tensor     = ! <== .TRUE. for polarizability calcs; Embedded Finite Field Polarizability  
  
GaussianCube    = ! <== .TRUE. for generating cube files for MO visualization  
GaussianCube_step = ! <== time step for saving Gaussian Cube files (Integer)  
  
NetCharge       = ! <== .TRUE. for dumping charge Occupancy in pdb format  
CH_and_DP_step   = ! <== time step for saving charge and Induced DP values (Integer)  
                   ! <== pdb format: charge --> Occupancy ; DP --> next to occupancy  
  
DensityMatrix    = ! <== .TRUE. for generating data for postprocessing with manipulate program  
AutoCorrelation  = ! <== .TRUE. for generating data for postprocessing with manipulate program  
VDOS_           = ! <== .TRUE. velocity DOS data for postprocessing with manipulate program  
!-----  
!-----  
!-----  
!----- POTENTIALS  
!  
EnvField_       = ! <== .TRUE. for using electrostaic Potential produced by Environment ; default = .FALSE.  
Environ_Type = ! <== choose from { Ch_MM , DP_QM , DP_MM };  
                 ! <== Ch_MM = point charges ; dipoles: { DP_QM , DP_MM } ...  
                 ! <== DP_MM = dipole moment from classical point charges  
                 ! <== DP_QM = dipole moment from quantum MO  
  
Environ_step = ! <== step for updating EnvField (Integer) ; default = 5  
  
Coulomb_        = ! <== .TRUE. for dipole potential for solvent molecules ; default = .FALSE.  
  
Induced_        = ! <== .TRUE. for induced dipole potential
```

# card.inpt

- Check the file “\$DYNEMOLDIR/card\_file\_formats” for guidance

For instance:

```
|-----  
|      DOS calculations  
|  
| sigma =           ! <== Gaussian broadening of DOS peaks in eV (Real) ; default = 0.04  
|  
| DOS_range = real_interval( min , max ) ! <== (min,max) Real values; defines energy range of DOS calculations  
|-----  
|      SPECTRUM calculations  
|  
| occupied = real_interval( min , max ) ! <== (min,max) Real values; defines energy range of occupied MOs  
| empty    = real_interval( min , max ) ! <== (min,max) Real values; defines energy range of empty MOs
```

```
|-----  
|      Genetic_Alg and CG OPTIMIZATION parameters  
|  
| Pop_Size      = Integer          ! <== Population size of candidate solutions for Genetic-Algorithm (Integer)  
| N_generations = Integer          ! <== number of iterations (Integer)  
| Pop_range     = Real             ! <== range of variation of parameters [0:1] (Real)  
| selection_by   =                 ! <== option = {roullete,ranking,sorting}; fitness selection method  
| Mutation_rate = Real             ! <== range of variation of parameters [0:1] (Real)  
| Adaptive_      = Logical          ! <== true -> Adaptive GA method; gradually tights cost function on-the-fly (Logical)  
| Mutate_Cross   = Logical          ! <== false -> pure Genetic Algorithm ; prefer false for fine tuning! (Logical)  
| CG_            = Logical          ! <== for using CONJUGATE GRADIENT method on Top_Selection after genetic algorithm (Logical)  
| Top_Selection  = Integer          ! <== top selection to undergo CG  
| profiling      = Logical          ! <== generates analysis of the optimization process
```

# card.inpt

- Check the file “\$DYNEMOLDIR/card\_file\_formats” for guidance

For instance:

```
!-----  
| MOLECULAR MECHANICS parameters  
|-----  
|  
| SYSTEM INFO  
|  
| N_of_molecules = ! <== total number of molecules (Integer)  
| N_of_species = ! <== total number of species (Integer)  
|-----  
! repeat the following information filling for all the different species ...  
attention: KEEP FORMAT  
!  
species(1) % residue = ! <== Residue label for species 1 ; character(len3)  
species(1) % N_of_molecules = ! <== Number of molecules of species (Integer)  
species(1) % N_of_atoms = ! <== Number of atoms comprising a single molecule of species i (Integer)  
species(1) % flex = ! <== .TRUE. for Flexible ; .FALSE. for rigid  
  
species(2) % residue = ! <== Residue label for species 2 ; character(len3)  
species(2) % N_of_molecules = ! <== Number of molecules of species (Integer)  
species(2) % N_of_atoms = ! <== Number of atoms comprising a single molecule of species i (Integer)  
species(2) % flex = ! <== .TRUE. for Flexible ; .FALSE. for rigid  
. . .  
species(n) % residue = ! <== Residue label for species n ; character(len3)  
species(n) % N_of_molecules = ! <== Number of molecules of species (Integer)  
species(n) % N_of_atoms = ! <== Number of atoms comprising a single molecule of species i (Integer)  
species(n) % flex = ! <== .TRUE. for Flexible ; .FALSE. for rigid  
  
Selective_Dynamics = ! <== .TRUE. for ad_hoc_MM_tuning of MegaMass to selected atoms ; default = .FALSE.
```

# card.inpt

- Check the file “\$DYNEMOLDIR/card\_file\_formats” for guidance

For instance:

```
!-----  
| ENVIRONMENT parameters  
|  
thermostat =           ! <== choose from { Berendsen, Nose_Hoover, Microcanonical }  
  
temperature =          ! <== Bath Temperature (K) , (Real) ; default = 300  
pressure =             ! <== Pressure in atm , (Real) ; default = 1  
  
thermal_relaxation_time = ! <== Temperature coupling term with the bath (Real)  
                           ! <== SMALL = STRONG coupling ; use "infty" to decouple  
                           ! <== picosecond ; default 0.25  
  
pressure_relaxation_time = ! <== Pressure coupling term (Real)  
                           ! <== SMALL = STRONG coupling ; use "infty" to decouple  
                           ! <== picosecond ; default = infty  
  
cutoff_radius =        ! <== Cut off radius (Angs.) for electrostatic and LJ interactions (Real) ; default = 50.  
damping_Wolf =         ! <== damping parameter (Angs.^{-1}) ; default value = 0.001  
                       ! <== Wolf's method damping parameter (length^{-1}) ; (J. Chem. Phys. 1999; 110(17):8254)  
                       ! <== relevant quantity: R_c*Wolf ~ ....  
!  
| EXECUTION INFO  
|  
driver_MM      =           ! <== choose from { MM_Dynamics , MM_Optimize , NormalModes , Parametrize }  
  
read_velocities =          ! <== .TRUE. for reading the initial velocities : T_ , F_ ; default = .TRUE.  
  
MM_input_format =          ! <== choose from { GMX, NAMD, GAFF } ; GMX = OPLS , GAFF and NAMD = Amber  
  
MM_log_step     =          ! <== step for saving MM results & parameters (Integer) ; default = 50  
  
MM_frame_step   =          ! <== step for saving MM results & parameters (Integer) ; default = 50  
  
Units_MM       =          ! <== choose OUTPUT energy units: eV or kj-mol ; default = eV
```

# card.inpt

- Check the file “\$DYNEMOLDIR/card\_file\_formats” for guidance

For instance:

```
!-----  
!  
!-----  
AD-HOC settings  
(ad_hoc = true)  
  
ad_hoc:QM_MM:feature(start:end) = fixing           ! <== QM_MM = QM or MM (apply settings to either realm)  
ad_hoc:QM_MM:feature(start:end) = fixing           ! <== feature = {residue , nr , fragment , V_shift , etc ...}; check types  
ad_hoc:QM_MM:feature(start:end) = fixing           ! <== (start:end) range of change, end >= start; atom index  
ad_hoc:QM_MM:feature(start:end) = fixing           ! <== fixing, depends on feature: 3-letter label, int_value, or real_value  
.  
.  
.  
!  
!-----
```

# Running Dynemol

In directory \$DYNEMOLWORKDIR:

- have the appropriate input files for the job:

(\* mandatory)

- card.inpt \*
- input.pdb \*      <== system coordinates
- velocity.inpt
- force-field related files
- opt\_eht\_parms.inpt

# Running Dynemol

In directory \$DYNEMOLWORKDIR:

- the execution script run-SN.sh (basic model of script)

```
#!/bin/bash

#####
#Script Name    : run-SN.sh
#Description   : execute Dynemol from pwd using local dynemol src files
#Args          : none
#Author        : Luis G C Rego
#####
#!/bin/bash

export DYNEMOLWORKDIR=$(pwd)
export DYNEMOLDIR=/home/lrego/QMMM/development/dynemol-dir

$DYNEMOLDIR/dynemol
```

- On the terminal run  
> ./run-SN.sh

# Running Dynemol

In directory \$DYNEMOLWORKDIR:

- Outputs , results , log-files , security-files , etc. are stored in:
  - \$DYNEMOLWORKDIR/ancillary.trunk/
  - \$DYNEMOLWORKDIR/dos.trunk/
  - \$DYNEMOLWORKDIR/dyn.trunk/
  - \$DYNEMOLWORKDIR/log.trunk/
  - \$DYNEMOLWORKDIR/MO.trunk/
  - \$DYNEMOLWORKDIR/opt.trunk/
- *which are deleted and made anew at every execution.*
- *each directory gets a copy of card.inpt for future reference.*

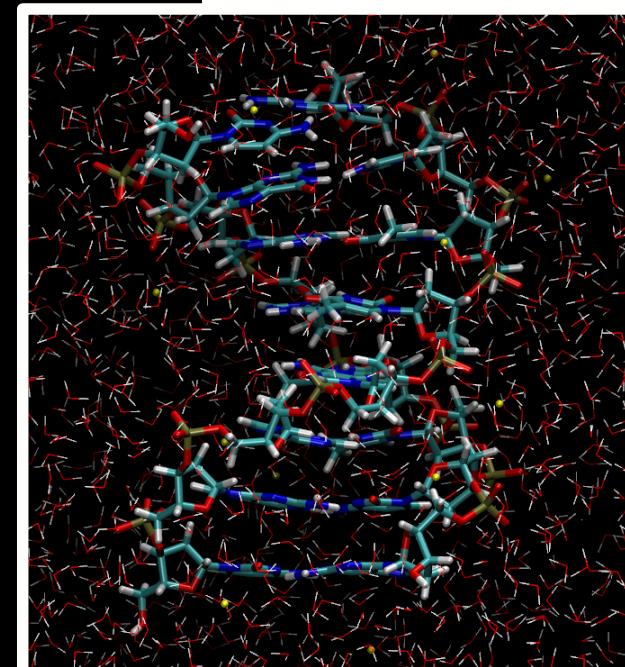
# Setting Up the System in card.inpt

Example: DNA strand in water + counter ions

```
!-----  
! Periodic Boundary Conditions  
  
PBC = [ 1 , 1 , 1 ]                                ! <== PBC replicas : 1 = yes , 0 = no  
!  
!  
!-----  
SYSTEM INFO  
  
N_of_molecules = 2452                               ! <== total number of molecules  
N_of_species   = 3                                  ! <== total number of species  
  
species(1) % residue      = DNA                  ! <== Residue label for species 1  
species(1) % N_of_molecules = 1                    ! <== Number of molecules of species  
species(1) % N_of_atoms   = 506                  ! <== Number of atoms comprising a single molecule of species i  
species(1) % flex          = true                 ! <== .TRUE. for Flexible  
  
species(2) % residue      = Na+                  ! <== Residue label for species 2  
species(2) % N_of_molecules = 14  
species(2) % N_of_atoms   = 1  
species(2) % flex          = true  
  
species(3) % residue      = H2O                  ! <== Residue label for species n  
species(3) % N_of_molecules = 2437  
species(3) % N_of_atoms   = 3  
species(3) % flex          = true  
!  
!-----  
AD-HOC settings  
(ad_hoc = true)  
  
ad_hoc:QM:QMMM(507:520) = false  
ad_hoc:QM:QMMM(521:2452) = false  
!  
!  
EXECUTION INFO  
  
driver_MM = MM_Dynamics  
  
MM_input_format = GAFF                            ! <== choose from {GMX, NAMD, GAFF}; GMX = OPLS, GAFF and NAMD = Amber
```

Files:

- input.pdb
- DNA.psf
- Na+.psf
- H2O.psf
- input.prm



# Setting Up the System in card.inpt

Example: DNA strand in water + counter ions

H2O.psf

```
PSF

      1 !NTITLE
REMARKS segment TIP

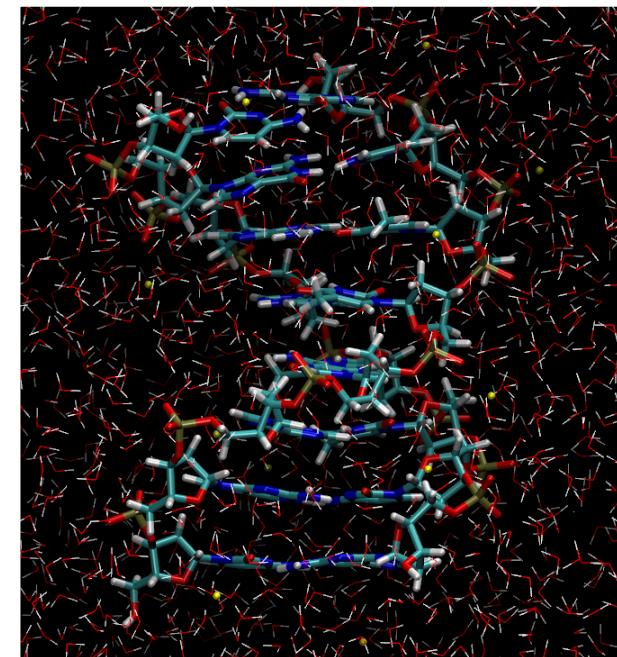
      3 !NATOM
      1 SYS      1       TIP3    OH2      OW      -0.834000    16.0000
      2 SYS      1       TIP3    H1       HW      0.417000     1.0080
      3 SYS      1       TIP3    H2       HW      0.417000     1.0080

      3 !NBOND: bonds
      2           1           3           1           3           2

      0 !NTHETA: angles
      0 !NPHTA: dihedrals
      0 !NIMPHI: impropers
      0 !NDON: donors
      0 !NACC: acceptors
      0 !NNB
```

Files:

- input.pdb
- DNA.psf
- Na+.psf
- H2O.psf
- input.prm



# Photochemistry in Solution

## Example: Azobenzene Molecule in Ethanol

```
!----- ACTION flags -----!
DRIVER = slice_A0

QM MM      = true
nuclear_matter = MDynamics
file_type     = structure
file_format   = pdb

PBC = [ 1 , 1 , 1 ]

electron_state = AZO:35
hole_state     = AZO:33

t_f = 1.0d0          ! <== final time in PICoseconds
n_t = 100000

!----- SYSTEM INFO -----!
N_of_molecules = 119           ! <== total number of molecules
N_of_species   = 2             ! <== total number of species

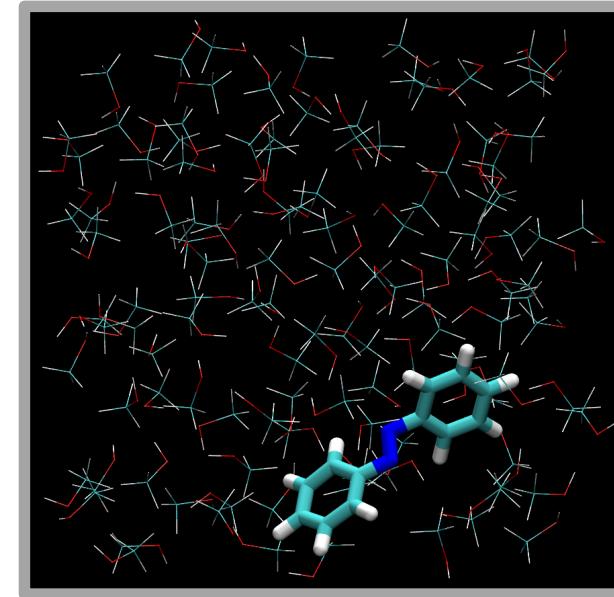
species(1) % residue       = COH    ! <== Residue label for species 1
species(1) % N_of_molecules = 118    ! <== Number of molecules of species 1
species(1) % N_of_atoms     = 6      ! <== # of atoms in a molecule of species 1
species(1) % flex           = true   ! <== Flexible : true , false

species(2) % residue       = AZO    ! <== Residue label for species 2
species(2) % N_of_molecules = 1
species(2) % N_of_atoms     = 24
species(2) % flex           = true

!----- AD-HOC settings -----!
OPT_parms = true
ad_hoc    = true
ad_hoc:QM:QMMM(1:708)=MM

!----- ENVIRONMENT parameters -----!
thermostat   = Microcanonical ! <== Berendsen, Nose_Hoover, Microcanonical
cutoff_radius = 9.d0           ! <== Cut off radius (Angs.) for LR interactions
damping_Wolf  = 0.0032d0       ! <== damping parameter (Angs.^-1)

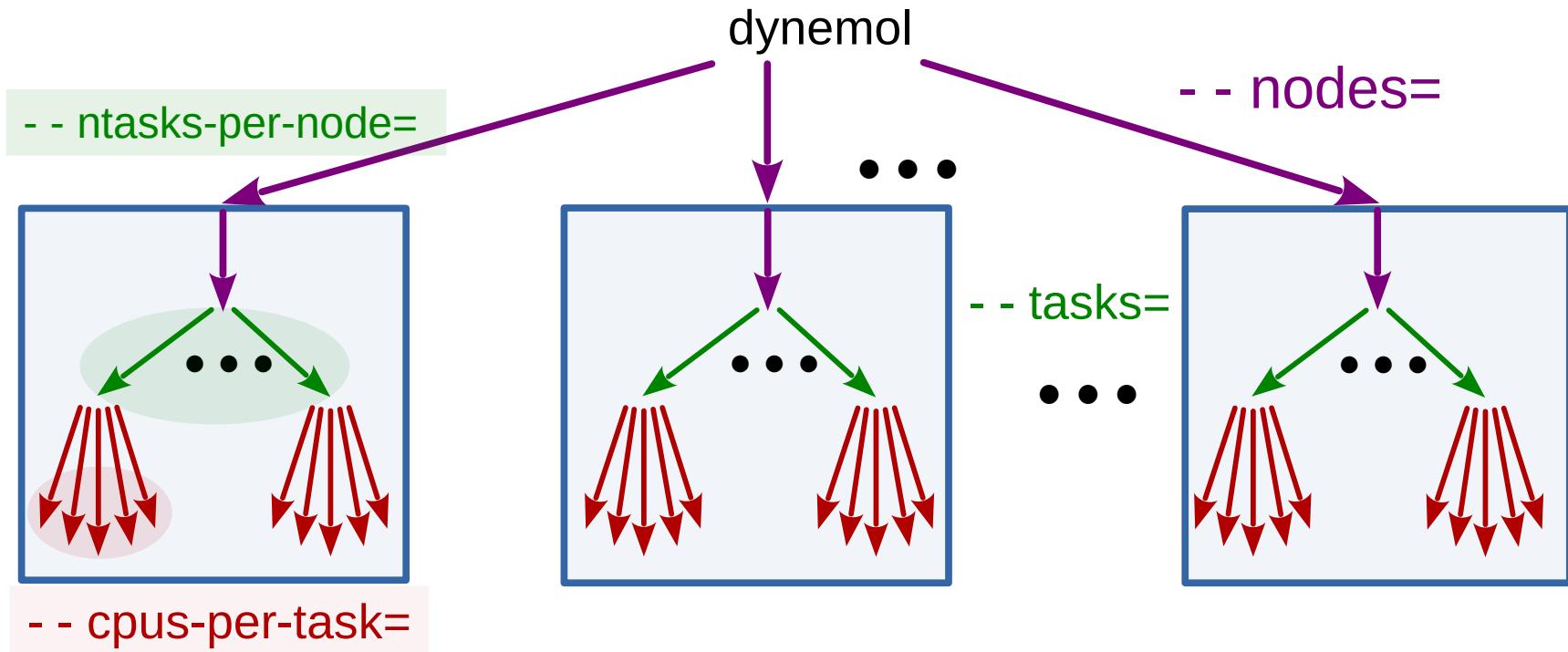
!----- GENERAL INFO -----!
driver_MM     = MM_Dynamics
read_velocities = true
MM_input_format = GMX
```



## Input Files:

- input.pdb
- AZO.itp
- COH.itp
- topol.top
- velocity.inpt
- opt\_eht\_parms.input

# Hybrid MPI + openMP mode on SLURM



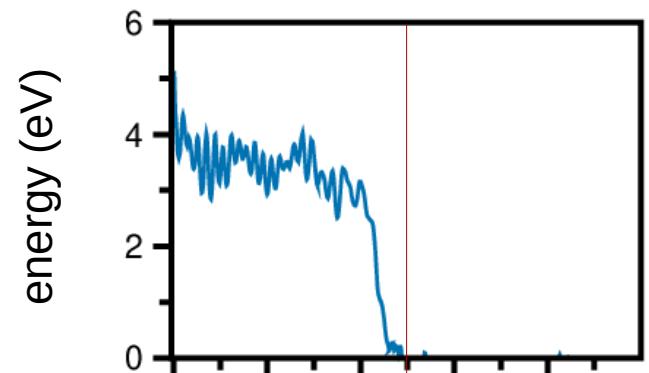
## Example: Azobenzene Molecule in Ethanol

# cpus (cores)	nodes	ntasks/node	ntasks	cpus/task	Time/(1000 iteration steps)
96	4	1	4	24	144 seconds
96	4	2	8	12	85.7 seconds
96	4	3	12	8	77.4 seconds

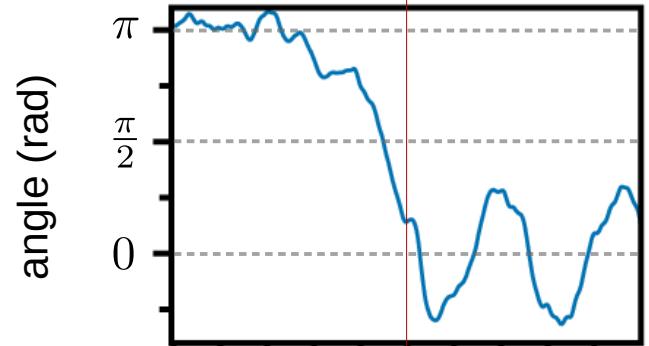
# Photochemistry in Solution

Example: Azobenzene Molecule in Ethanol

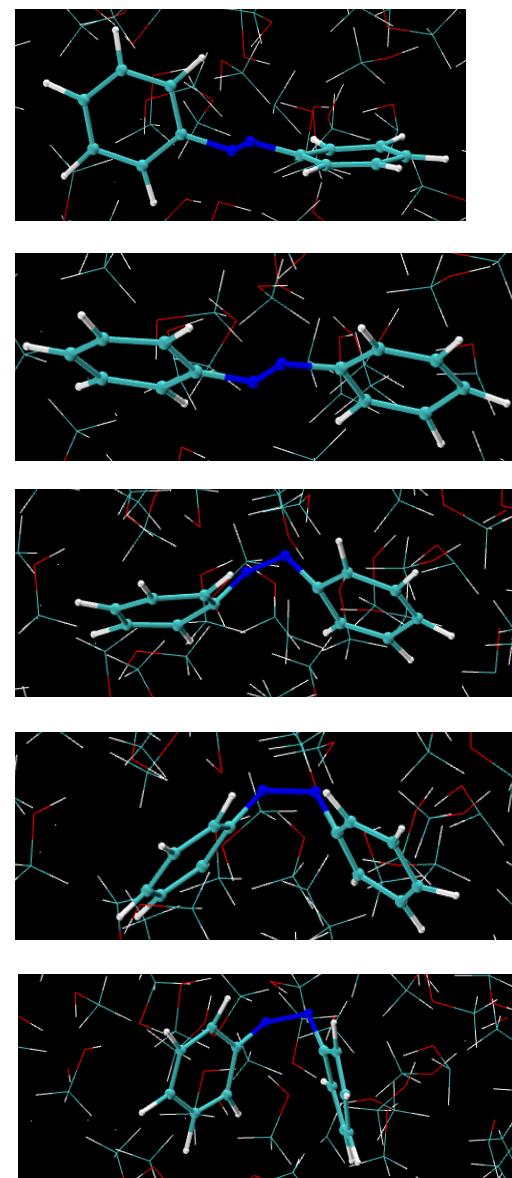
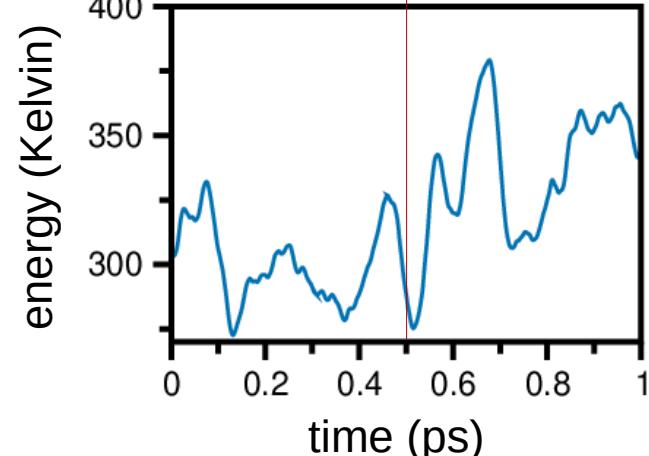
El - H<sub>1</sub> excitation energy



CNNC dihedral angle



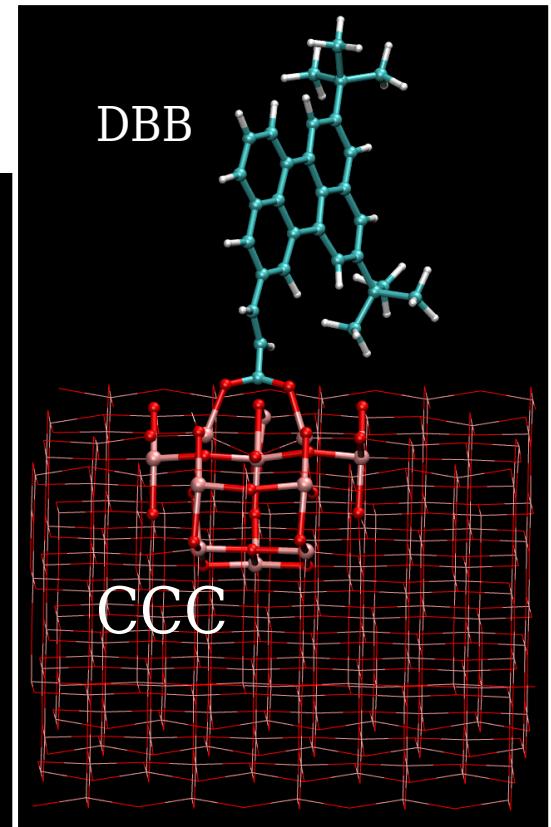
AZ0+EtOH  
 $E_{\text{kinetic}}$



# Setting Up the System in card.inpt

Example: Dye-sensitized semiconductor

```
!----- ACTION flags -----!  
DRIVER      = MM_Dynamics  
  
nuclear_matter = MDynamics  
file_type      = structure          ! <== structure or trajectory  
file_format     = pdb               ! <== xyz , pdb or vasp  
  
PBC = [ 1 , 1 , 0 ]  
  
t_f  = 50.0                      ! <== final time in PICOseconds  
n_t  = 100000                     ! <== number of time steps  
  
!----- SYSTEM INFO -----!  
  
N_of_molecules = 2                ! <== total number of molecules  
N_of_species   = 2                ! <== total number of species  
  
species(1) % residue      = DBB  ! <== Residue label for species 1  
species(1) % N_of_molecules = 1    ! <== Number of molecules of species i  
species(1) % N_of_atoms    = 64   ! <== # of atoms comprising a single molecule of species i  
species(1) % flex          = true ! <== Flexible : true , false  
  
species(2) % residue      = CCC  ! <== Residue label for species 2  
species(2) % N_of_molecules = 1    !  
species(2) % N_of_atoms    = 479  !  
species(2) % flex          = false
```



Files:

- input.pdb
- DBB.psf
- CCC.psf
- input.prm

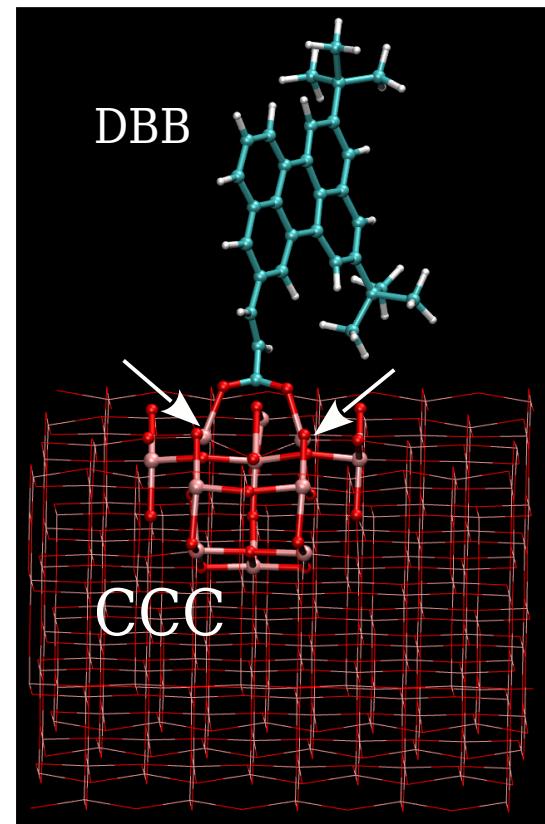
# Setting Up the System in card.inpt

Example: Dye-sensitized semiconductor

DBB.psf

```
1 !NTITLE
REMARKS DBB=Perylene_Dye_typeB_tert-Butyl

64 !NATOM
 1 SYS    1      DBB    CA     CA      0.108060   12.011
 2 SYS    1      DBB    CA     CA     -0.256800   12.011
 3 SYS    1      DBB    CA     CB      0.015126   12.011
 .
 .
 .
 .
 .
 .
 61 SYS   1      DBB    HC     HC      0.041699   1.008
 62 SYS   1      DBB    HC     HC      0.038063   1.008
 63 SYS   1      DBB    TI     TI      2.1960    47.8671
 64 SYS   1      DBB    TI     TI      2.1960    47.8671
```



Files:

- input.pdb
- DBB.psf
- CCC.psf
- input.prm

# Setting Up the System in card.inpt

Example: Dye-sensitized semiconductor

CCC.psf

```
1 !NTITLE
REMARKS CCC=TiO2 cluster

479!NATOM
1   SYS    1   CCC    TI    TI    2.1960  47.8671
2   SYS    1   CCC    TI    TI    2.1960  47.8671
.
.
.
.
.
.
478   SYS    1   CCC    0     0    -1.0980  15.9994
479   SYS    1   CCC    0     0    -1.0980  15.9994
```

0 !NBOND: bonds

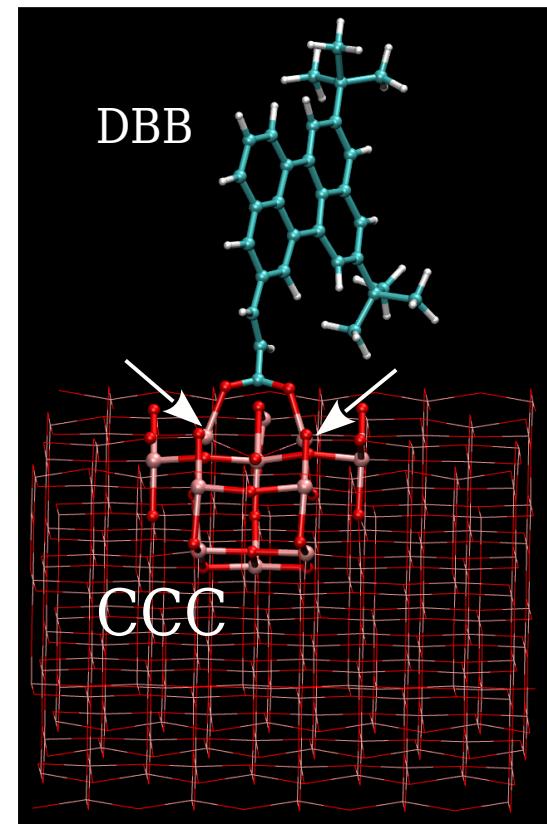
0 !NTHETA: angles

0 !NPHI: dihedrals

0 !NIMPHI: impropers

31 !AD-HOC: flex

```
74    TI  true
136   0   true
204   0   true
71    TI  true
192   0   true
132   0   true
78    TI  true
326   0   true
266   0   true
.
.
.
.
.
.
111   TI  true
424   0   true
265   0   true
```



Files:

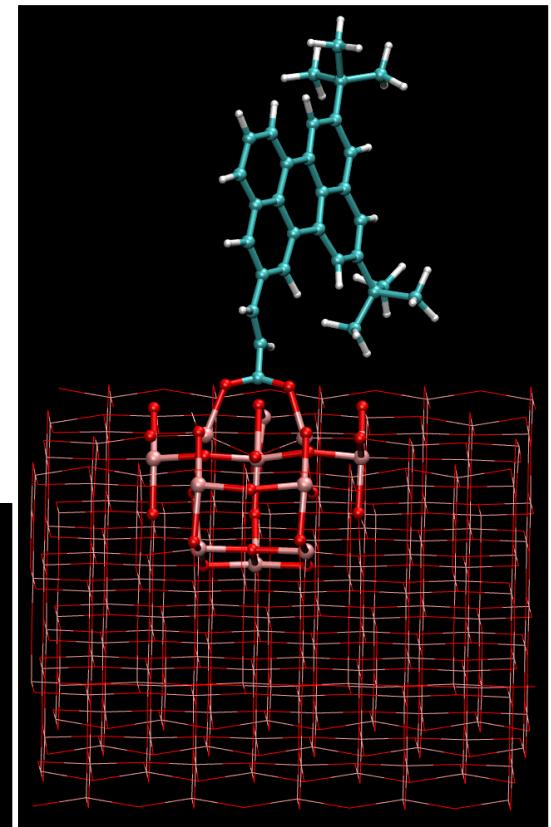
- input.pdb
- DBB.psf
- CCC.psf
- input.prm

# Setting Up the System in card.inpt

Example: Dye-sensitized semiconductor

Files:

- input.pdb
- DBB.psf
- CCC.psf
- input.prm



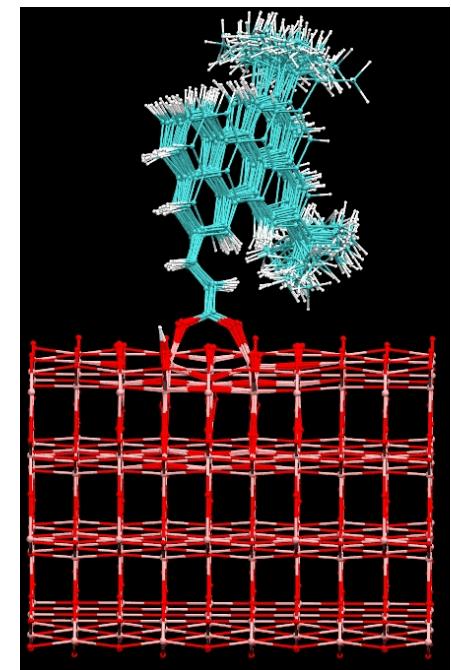
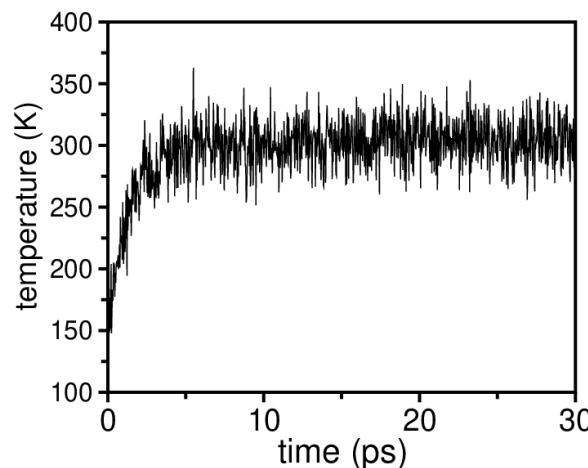
```
|-----  
|           ENVIRONMENT parameters ...  
|  
thermostat      = Microcanonical ! <== Berendsen, Nose_Hoover, Microcanonical  
temperature       = 300.d0          ! <== Bath Temperature (K)  
thermal_relaxation_time = 5.d-1    ! <== Temperature coupling term with the bath  
                                    ! <== SMALL = STRONG ; use "= infinity" to decouple  
cutoff_radius     = 50.d0          ! <== Cut off radius (Angs.) for electrostatic and LJ interactions  
damping_Wolf      = 0.0005         ! <== damping parameter (Angs.^-1)  
|-----  
|           GENERAL INFO ...  
|  
driver_MM         = MM_Dynamics   ! <== MM_Dynamics , MM_Optimize , NormalModes , Parametrize  
read_velocities   = true           ! <== reads the initial velocities : T_ , F_  
MM_input_format   = GAFF          ! <== GMX, NAMD, GAFF  
MM_log_step        = 50             ! <== step for saving MM results & parameters  
MM_frame_step      = 100            ! <== step for saving MM results & parameters
```

# Thermalization

## Example: Dye-sensitized semiconductor

```
thermostat      = Berendsen      ! <== Berendsen, Nose_Hoover, Microcanonical
temperature      = 300.d0        ! <== Bath Temperature (K)
thermal_relaxation_time = 7.d-1   ! <== Temperature coupling term with the bath
                                    ! <== SMALL = STRONG ; use "= infinity" to decouple
read_velocities  = false         ! <== reads the initial velocities : T_ , F_
```

> \$DYNEMOLDIR/dynemol



```
thermostat      = Microcanonical ! <== Berendsen, Nose_Hoover, Microcanonical
read_velocities = true          ! <== reads the initial velocities : T_ , F_
```

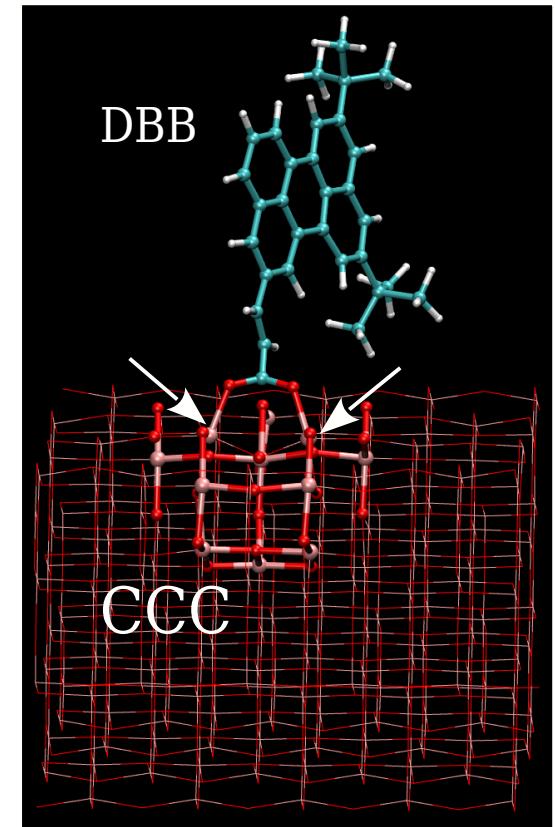
> mv velocity\_MM.out velocity\_MM.inpt  
> \$DYNEMOLDIR/dynemol resume

# Interfacial Electron Transfer

## Example: Dye-sensitized semiconductor

- Photoinduced electron transfer from DBB:LUMO → TiO<sub>2</sub> cluster
- Photoexcited electron to DBB:LUMO
- Hole in DBB:HOMO
- Atoms at fixed positions
- PBC in the (x,y) plane

```
!-----  
!           ACTION   flags  
  
!-----  
DRIVER = q_dynamics  
  
survival      = true  
nuclear_matter = extended_sys  
file_type      = structure          ! <= structure or trajectory  
file_format    = pdb               ! <= xyz , pdb or vasp  
  
PBC = [ 1 , 1 , 0 ]  
  
electron_state = DBB:84  
hole_state     = DBB:83  
  
t_f  =  0.5d0                      ! <= final time in PIC0oseconds  
n_t  =  1000  
!-----  
!  
!           AD-HOC settings  
  
!-----  
OPT_parms = true  
ad_hoc = true  
ad_hoc:QM:residue(63:64)=CCC  
ad_hoc:QM:nr(63:64)=2  
ad_hoc:QM:V_shift(1:62)=0.6  
!-----  
!  
!           DOS settings  
  
sigma      =  0.040d0                !  
DOS_range = real_interval( -15.d0 , 0.d0 )
```

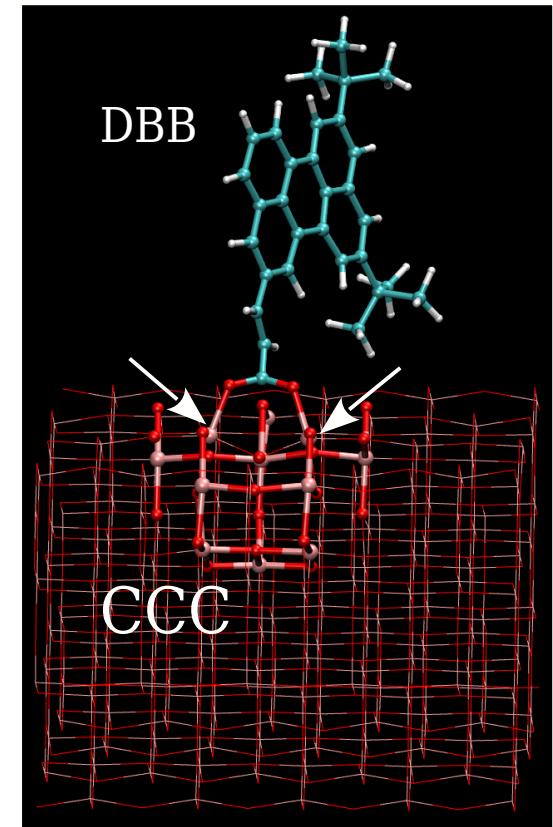


# Interfacial Electron Transfer

## Example: Dye-sensitized semiconductor

- Ad-hoc settings:
  - Dynemol reads opt\_eht\_parms.input
  - Ti atoms 63:64 belong to TiO<sub>2</sub> cluster (CCC)
  - Energy offset (V\_shift) of DBB fragment orbitals with respect to CCC

```
!----- ACTION flags -----!  
!  
DRIVER = q_dynamics  
  
survival      = true  
nuclear_matter = extended_sys  
file_type      = structure          ! <= structure or trajectory  
file_format     = pdb               ! <= xyz , pdb or vasp  
  
PBC = [ 1 , 1 , 0 ]  
  
electron_state = DBB:84  
hole_state     = DBB:83  
  
t_f  =  0.5d0                      ! <= final time in PIC0oseconds  
n_t  =  1000  
!  
!----- AD-HOC settings -----!  
  
OPT_parms = true  
ad_hoc = true  
ad_hoc:QM:residue(63:64)=CCC  
ad_hoc:QM:nr(63:64)=2  
ad_hoc:QM:V_shift(1:62)=0.6  
!  
!----- DOS settings -----!  
  
sigma      =  0.040d0                !  
DOS_range = real_interval( -15.d0 , 0.d0 )
```

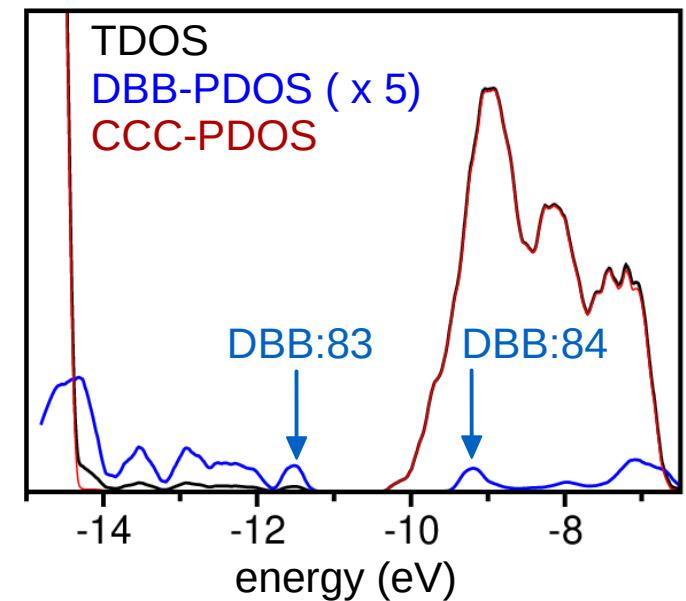
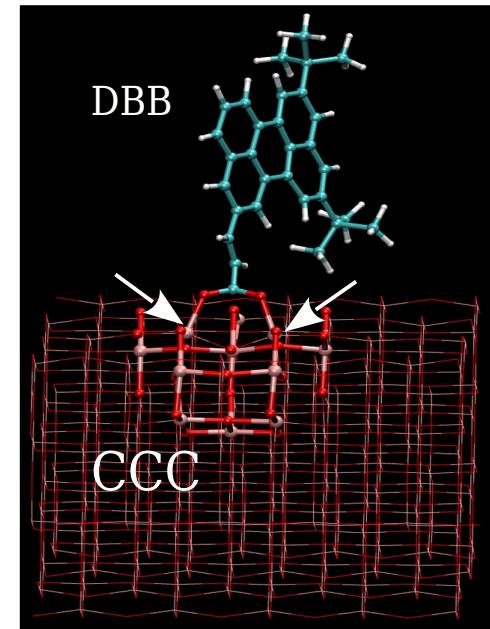


# Interfacial Electron Transfer

## Example: Dye-sensitized semiconductor

- Energy DOS settings:
  - calculates Total DOS (TDOS.dat)
  - calculates DOS projected on residues (“resname”-PDOS.dat)
  - results written in dos.trunk

```
!-----  
!           ACTION   flags  
  
! DRIVER = q_dynamics  
  
survival      = true  
nuclear_matter = extended_sys  
file_type      = structure          ! <= structure or trajectory  
file_format     = pdb               ! <= xyz , pdb or vasp  
  
PBC = [ 1 , 1 , 0 ]  
  
electron_state = DBB:84  
hole_state     = DBB:83  
  
t_f  =  0.5d0                      ! <= final time in PIC0oseconds  
n_t  =  1000  
!  
!           AD-HOC settings  
  
OPT_parms = true  
ad_hoc = true  
ad_hoc:QM:residue(63:64)=CCC  
ad_hoc:QM:nr(63:64)=2  
ad_hoc:QM:V_shift(1:62)=0.6  
!  
!           DOS settings  
  
sigma      = 0.040d0  
DOS_range = real_interval( -15.d0 , 0.d0 )  
!
```

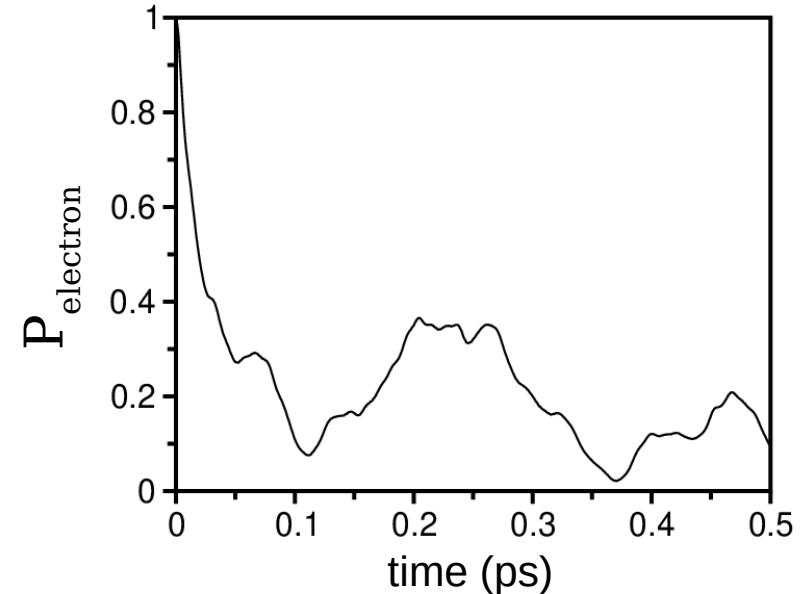
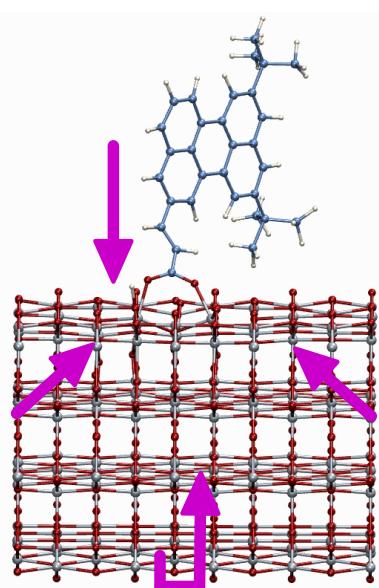


# Interfacial Electron Transfer

Example: Dye-sensitized semiconductor

## Finite-Size Spurious Artifacts

```
!-----  
          ACTION    flags  
!  
DRIVER = q_dynamics  
  
survival      = true  
nuclear_matter = extended_sys  
file_type     = structure      ! <== structure or trajectory  
file_format   = pdb           ! <== xyz , pdb or vasp  
  
PBC = [ 1 , 1 , 0 ]  
  
electron_state = DBB:84  
hole_state     = DBB:83
```



# Interfacial Electron Transfer

Example: Dye-sensitized semiconductor

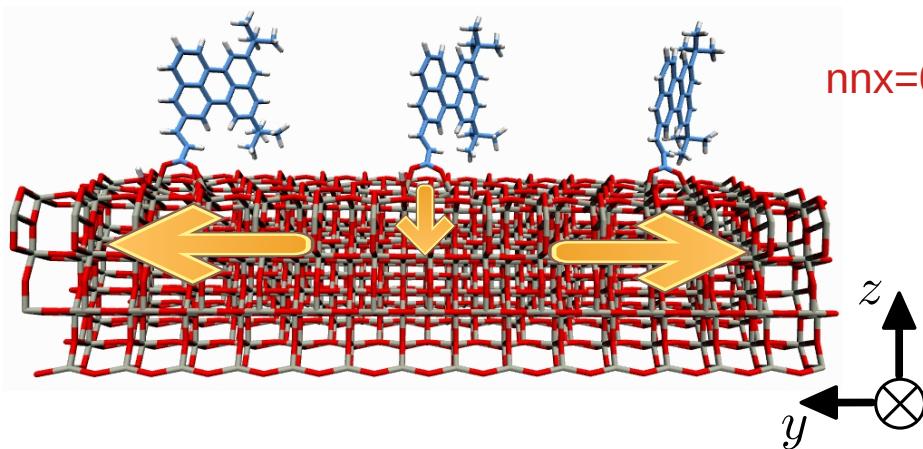
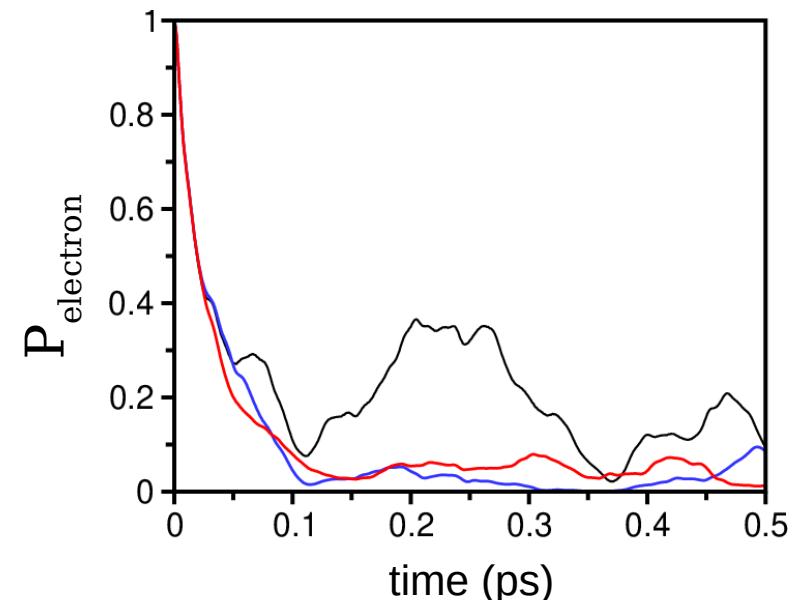
Nnx=0;nny=0

Finite-Size Spurious Artifacts

Nnx=1;nny=0

Nnx=0;nny=1

```
!-----  
!-----  
      ACTION   flags  
!-----  
  
DRIVER = q_dynamics  
  
survival      = true  
nuclear_matter = extended_sys  
file_type     = structure  
file_format    = pdb  
  
! <== structure or trajectory  
! <== xyz , pdb or vasp  
  
nnx = 0 ; nny = 1  
  
PBC = [ 1 , 1 , 0 ]  
  
electron_state = DBB:84  
hole_state     = DBB:83  
  
t_f  =  0.5d0  
n_t  =  1000  
  
! <== final time in PIC0seconds
```

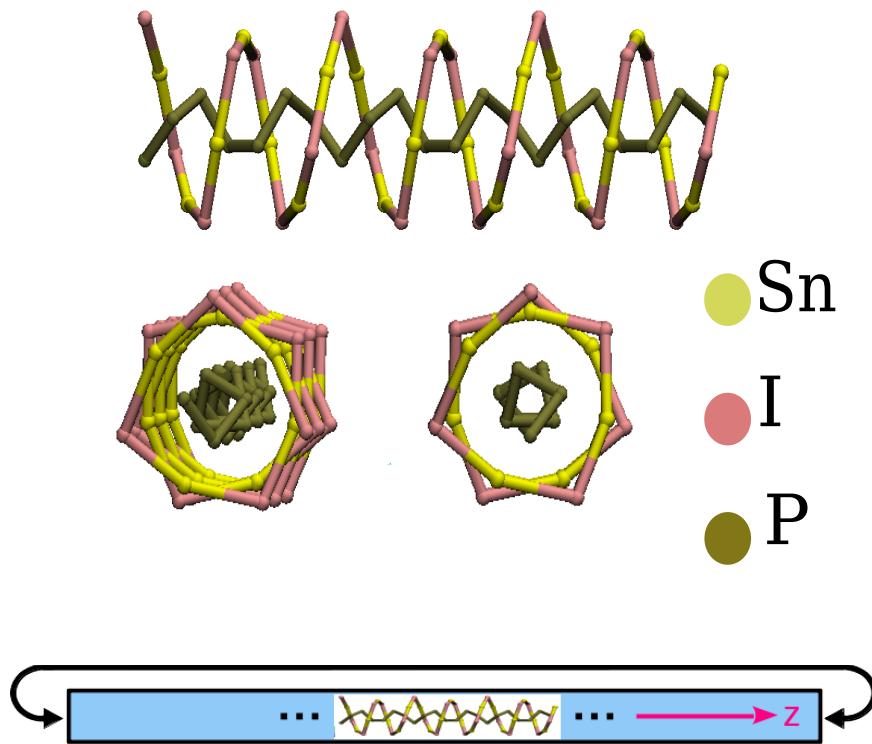


nmx=0,nny=1

Notice: not cost-effective

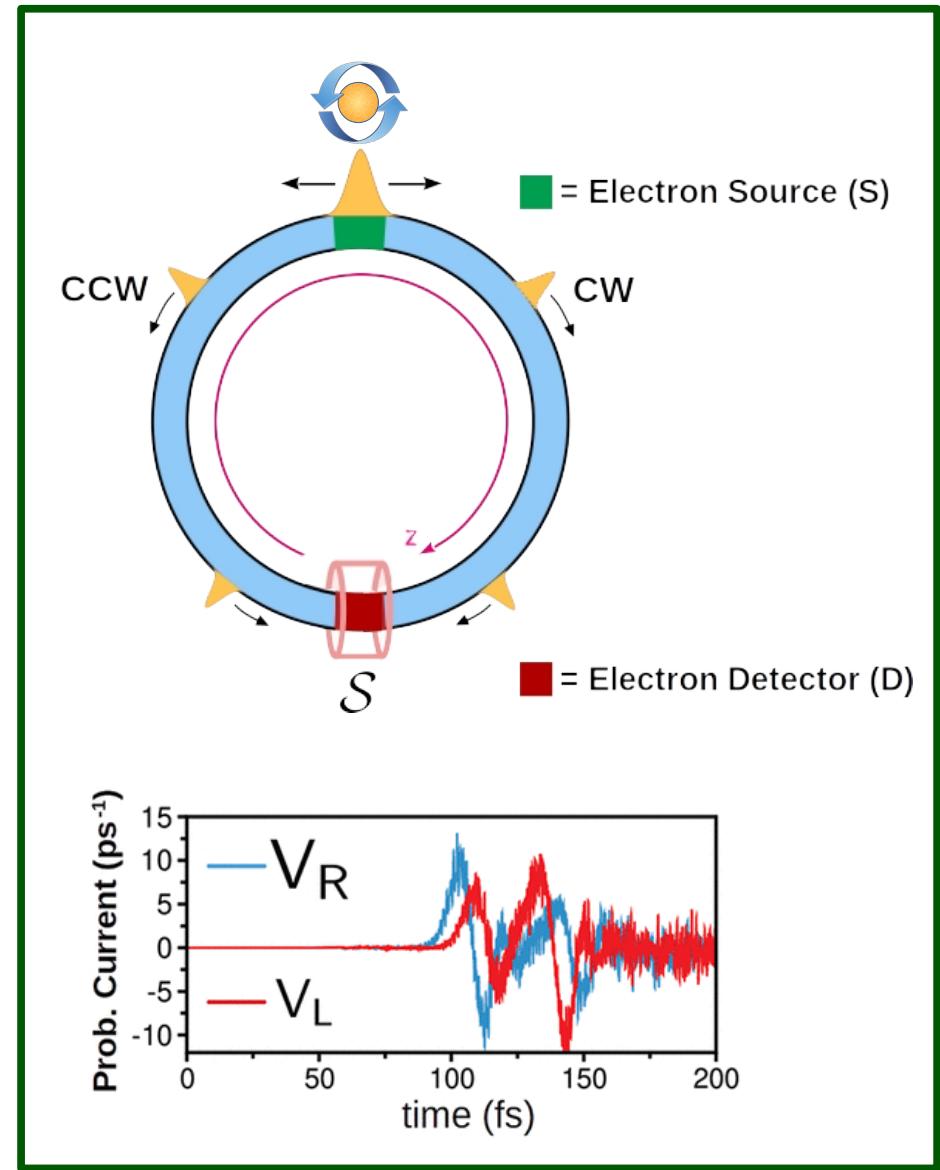
# Electron Propagation in Chiral nanowire

Inorganic Double Helices in Semiconducting SnIP



$$nnz = N_{\text{cell}}$$

$$\text{PBC} = [0, 0, 1]$$



# Setting Up the System in card.inpt

Example: Dye-sensitized semiconductor

Orbital Rendering of Charge Dynamics

```
!----- ACTION flags
! DRIVER = q_dynamics

survival      = true
nuclear_matter = extended_sys
file_type     = structure          ! <== structure or trajectory
file_format    = pdb               ! <== xyz , pdb or vasp

nnx = 0 ; nny = 0

PBC = [ 1 , 1 , 0 ]

electron_state = DBB:79
hole_state     = DBB:78

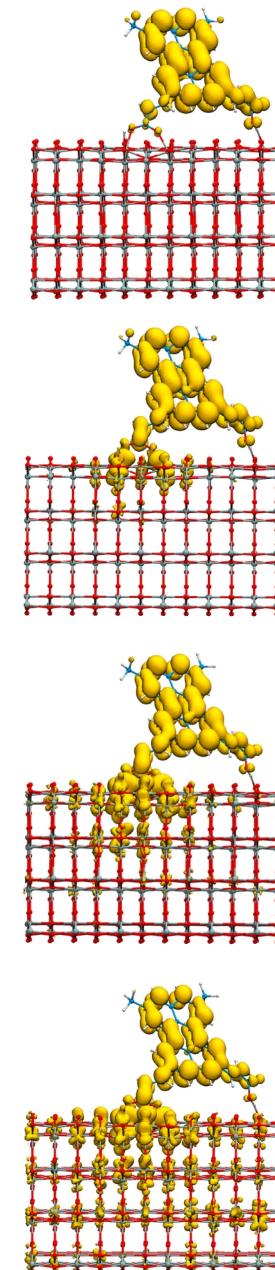
t_f  =  0.2d0                      ! <== final time in PICoseconds
n_t  =  200

!----- AD-HOC settings

OPT_parms = true
ad_hoc = true
ad_hoc:QM:residue(52:53)=CCC
ad_hoc:QM:nr(52:53)=2
ad_hoc:QM:V_shift(1:51)=0.6

!----- VISUALIZATION flags

GaussianCube    =  true   ! <== generating cube files for MO visualization
GaussianCube_step =  40    ! <== time step for saving Gaussian Cube files
```



# Interfacial Electron Transfer

Example: Dye-sensitized semiconductor

Average over conformations

```
!----- ACTION flags
DRIVER = avrg_configs

survival      = true
nuclear_matter = extended_sys
file_format    = pdb          ! <= xyz , pdb or vasp
file_type      = trajectory   ! <= structure or trajectory
frame_step     = 1            ! <= step for avrg_configs ;
                             frame_step <= size(trj) ; default = 1

nnx = 1 ; nny = 0

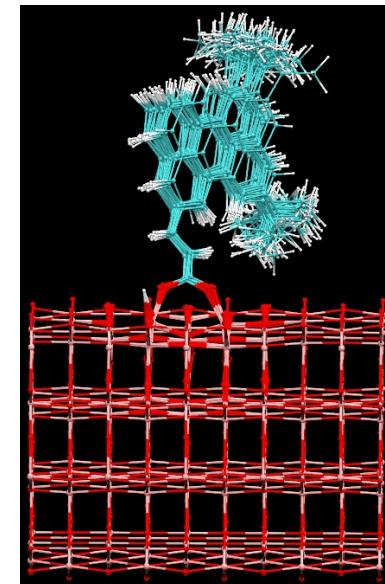
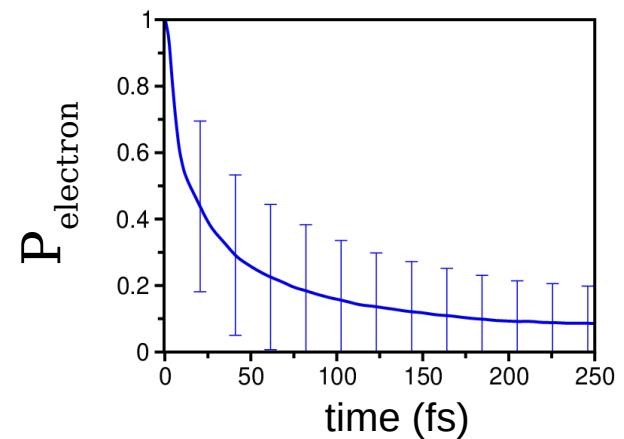
PBC = [ 1 , 1 , 0 ]

electron_state = DBB:84
hole_state     = DBB:83

t_f  =  0.5d0           ! <= final time in PICoseconds
n_t  =  1000

!----- AD-HOC settings
OPT_parms = true
ad_hoc = true
ad_hoc:QM:residue(63:64)=CCC
ad_hoc:QM:nr(63:64)=2
ad_hoc:QM:V_shift(1:62)=0.6

!----- DOS settings
sigma      = 0.040d0
DOS_range = real_interval( -15.d0 , 0.d0 )
```



Input File:

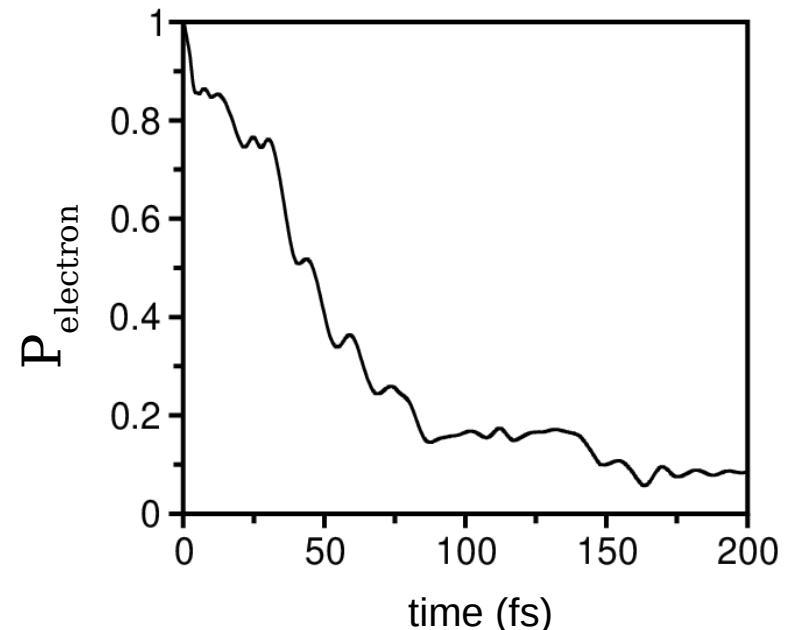
- frames.pdb

# Interfacial Electron Transfer

Example: Dye-sensitized semiconductor

Non-adiabatic Electron Transfer Dynamics

```
!-----  
!           ACTION   flags  
!<== Ehrenfest QMMM  
DRIVER      = slice_A0  
QMMM        = true  
survival    = true  
  
nuclear_matter = MDynamics  
file_type     = structure  
file_format   = pdb  
  
PBC = [ 1 , 1 , 0 ]  
  
electron_state = DBB:66  
hole_state     = DBB:65  
  
t_f  =  0.20          ! <== final time in PICoseconds  
n_t  = 10000  
!  
!-----  
!           SYSTEM INFO  
N_of_molecules = 2          ! <== total number of molecules  
N_of_species   = 2          ! <== total number of species  
  
species(1) % residue       = DBB  ! <== Residue label for species 1  
species(1) % N_of_molecules = 1  ! <== # of molecules of species 1  
species(1) % N_of_atoms    = 45  ! <== # of atoms in a molecule of species 1  
species(1) % flex           = true ! <== Flexible : true , false  
  
species(2) % residue       = CCC ! <== Residue label for species 2  
species(2) % N_of_molecules = 1  
species(2) % N_of_atoms    = 479  
species(2) % flex           = false  
!  
!           AD-HOC settings  
OPT_parms = true  
ad_hoc = true  
ad_hoc:QM:residue(45:46)=CCC  
ad_hoc:QM:nr(45:46)=2  
ad_hoc:QM:V_shift(1:44)=0.6
```



# Interfacial Electron Transfer

Example: Dye-sensitized semiconductor

Non-adiabatic Electron Transfer Dynamics

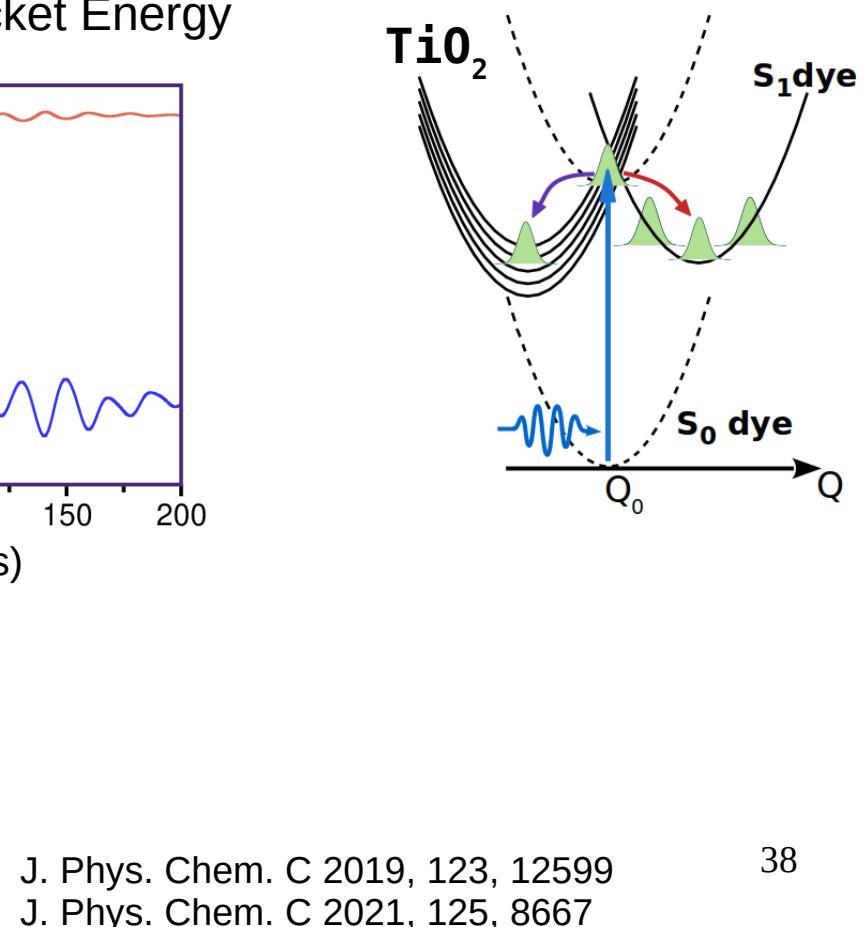
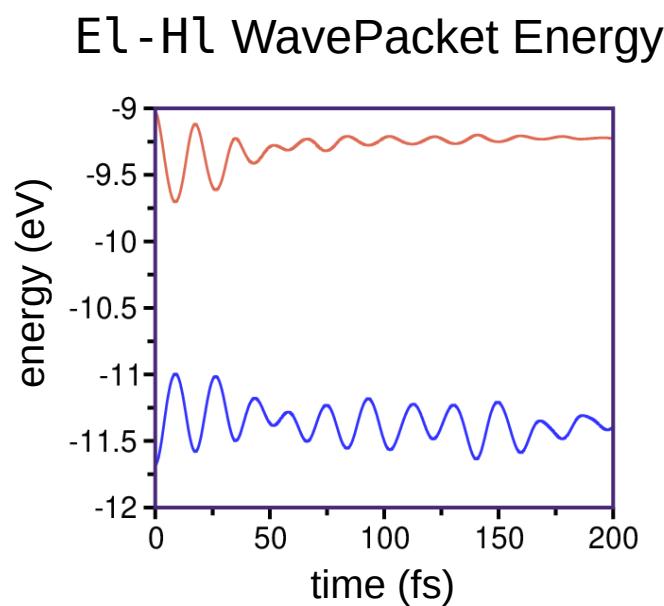
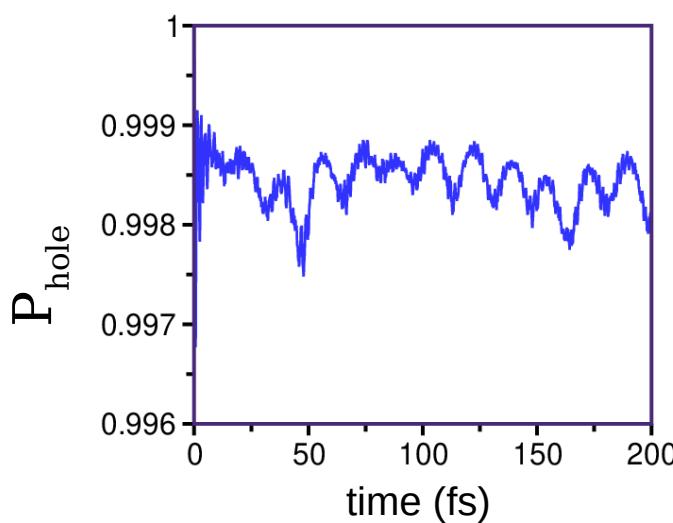
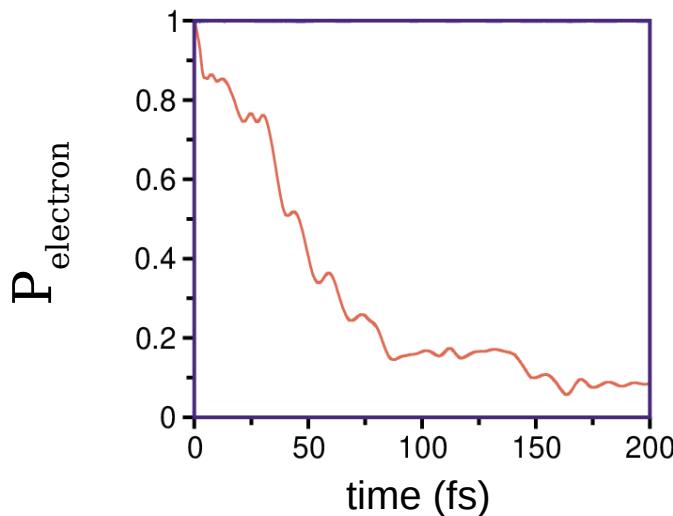
```
!-----  
!           ENVIRONMENT parameters  
  
thermostat      = Microcanonical ! <== Berendsen, Nose_Hoover, Microcanonical  
  
cutoff_radius   = 50.d0          ! <== Cut off radius (Angs.) for electrostatic  
damping_Wolf    = 0.0005         ! <== damping parameter (Angs.^-1)  
  
driver_MM       = MM_Dynamics   ! <== MM_Dynamics , MM_Optimize , NormalModes , Parametrize  
  
read_velocities = true  
  
MM_input_format = GAFF          ! <== GMX, NAMD, GAFF
```

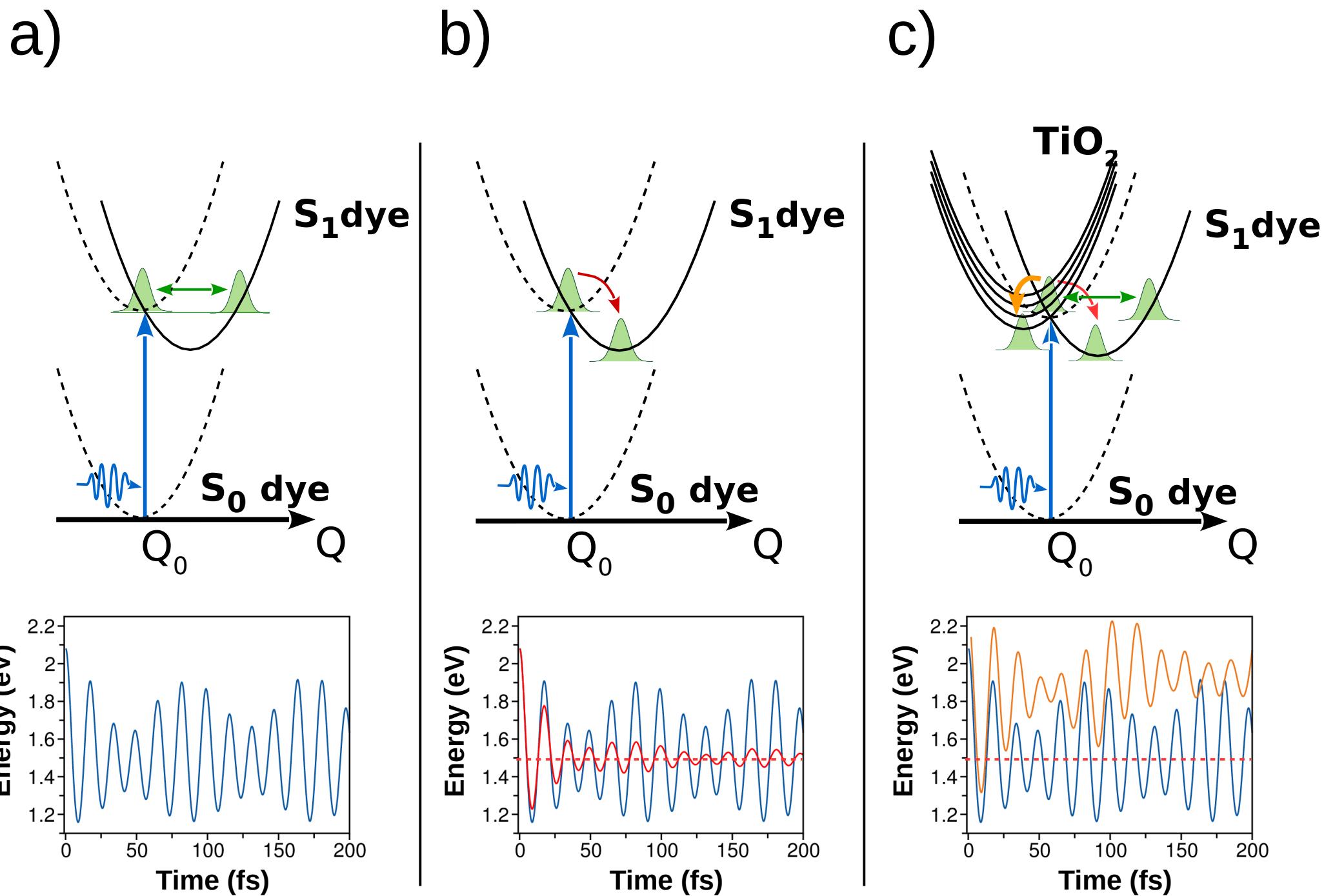


# Interfacial Electron Transfer

Example: Dye-sensitized semiconductor

Non-adiabatic Electron Transfer Dynamics



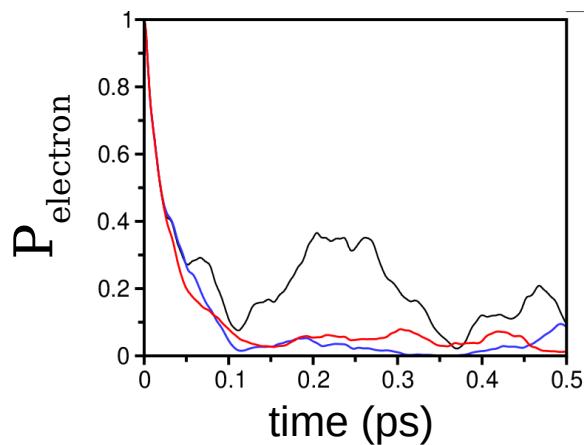


# Interfacial Electron Transfer

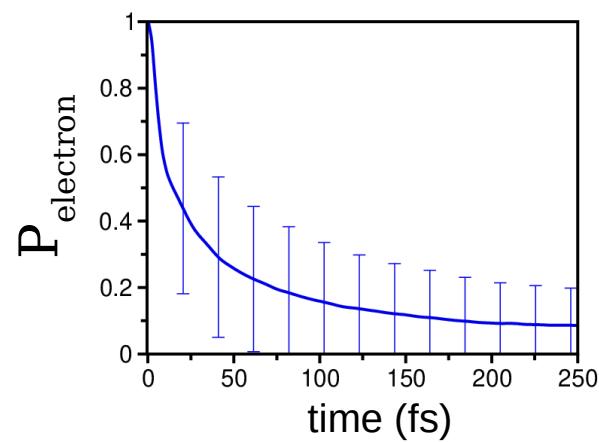
Example: Dye-sensitized semiconductor

Non-adiabatic Electron Transfer Dynamics on **rigid** structures

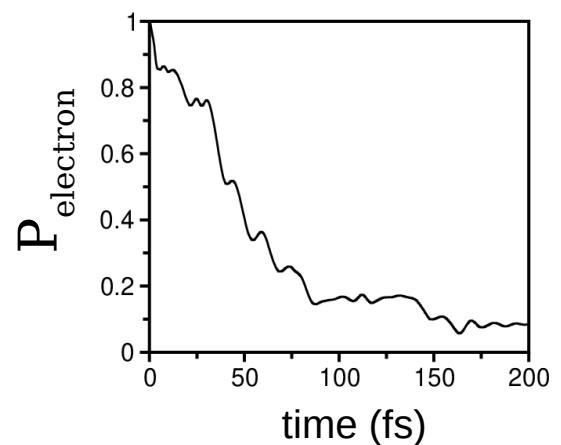
DRIVER = q\_dynamics



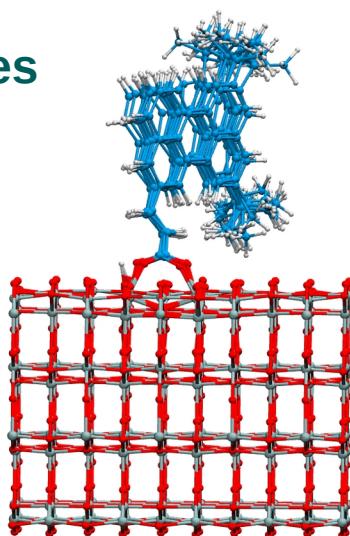
avrg\_configs



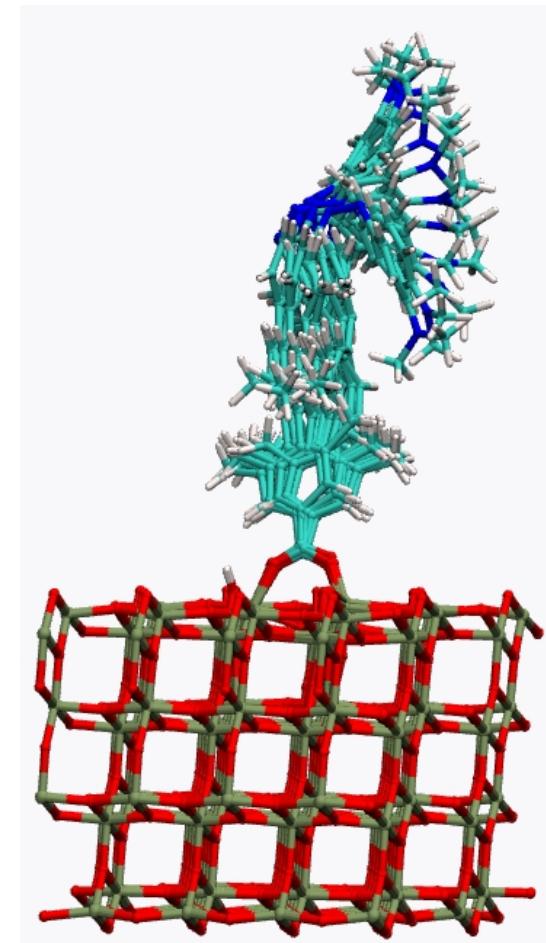
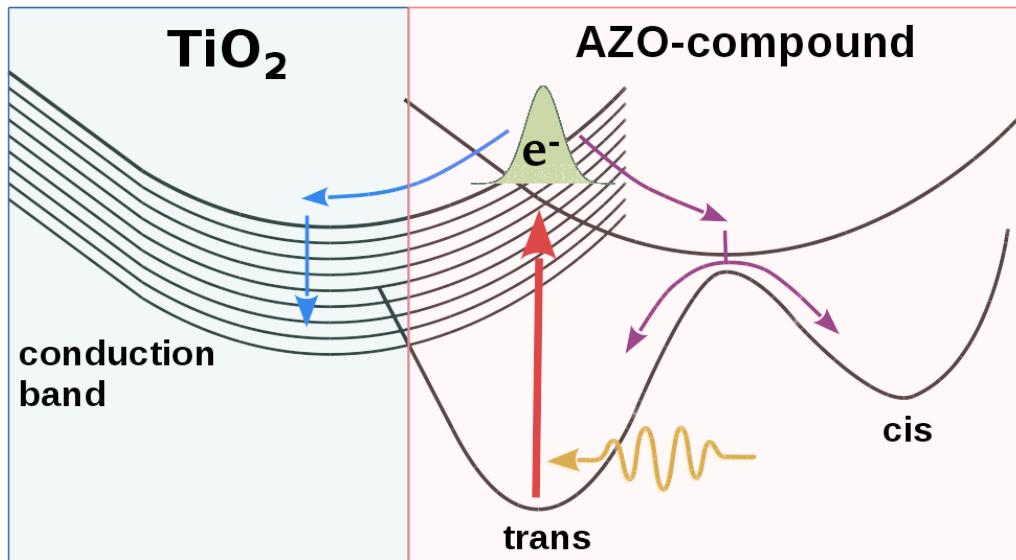
Ehrenfest



Similar behavior for rigid structures



# Charge Transfer *vs* Structural Relaxation



J. Phys. Chem. Lett. 2018, 9, 5926.  
J. Phys. Chem. C 2019, 123, 5692.  
J. Phys. Chem. Lett. 2015, 6, 2393.

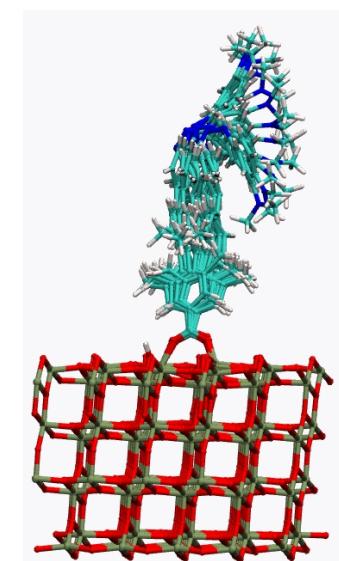
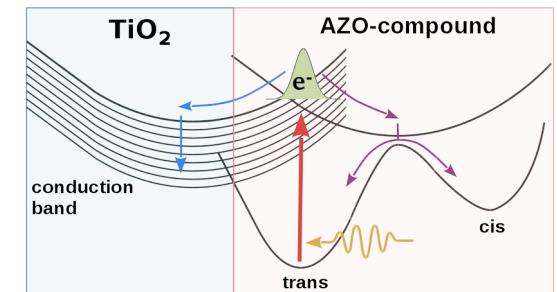
# Charge Transfer vs Structural Relaxation

## Sample card.inpt

```

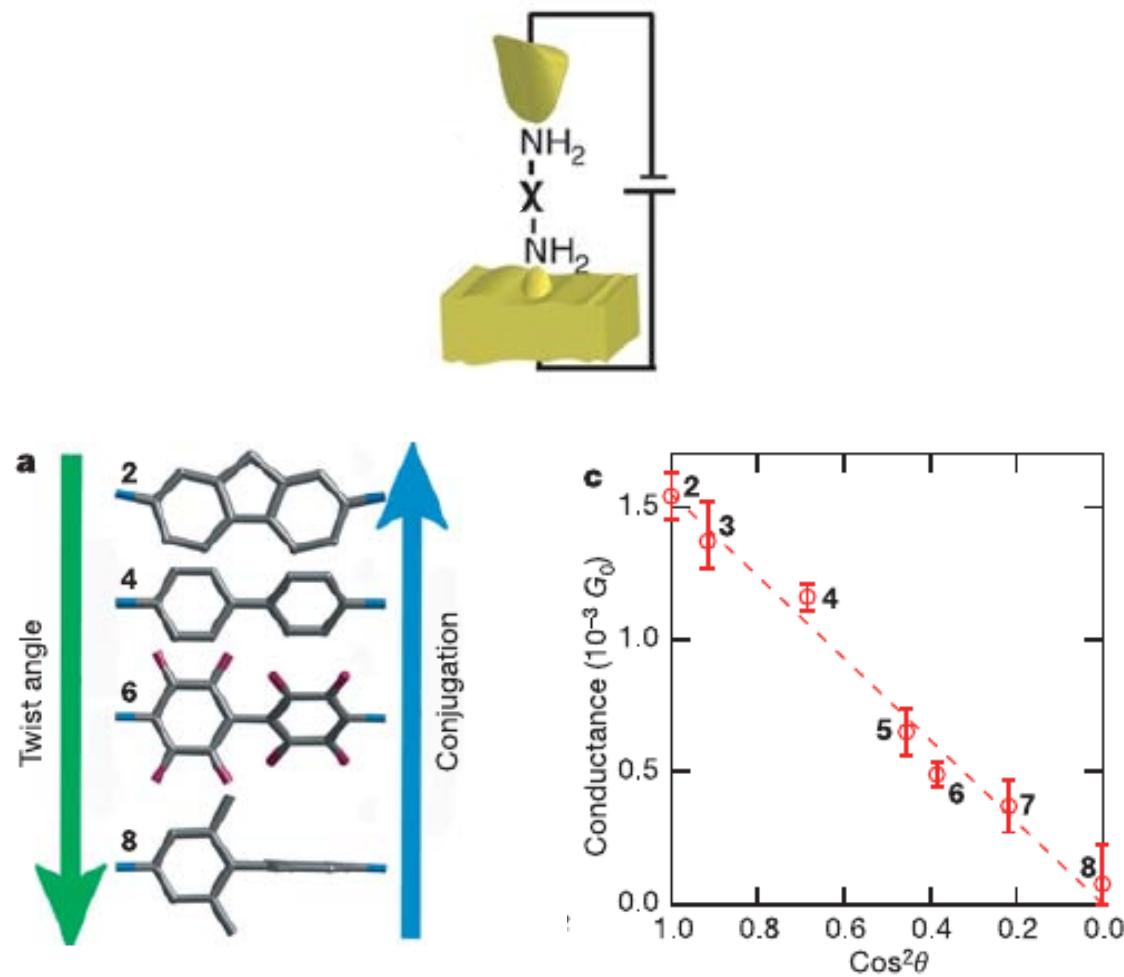
!
!----- ACTION flags
!----- DRIVER = slice_Cheb           ! <== slice_[Cheb, AO, FSSH, CSDM]
!----- QMMM      = true
!----- survival   = true
!----- nuclear_matter = MDynamics
!----- file_type   = structure          ! <== structure or trajectory
!----- file_format  = pdb               ! <== xyz , pdb or vasp
!----- PBC = [ 1 , 1 , 0 ]
!----- electron_state = AZD:92
!----- hole_state    = AZD:89
!----- t_f = 1.50                      ! <== final time in PIC0oseconds
!----- n_t = 75000
!
!----- SYSTEM INFO
!----- N_of_molecules = 2             ! <== total number of molecules
!----- N_of_species   = 2             ! <== total number of species
!----- species(1) % residue       = AZD      ! <== Residue label for species 1
!----- species(1) % N_of_molecules = 1      ! <== # of molecules of species 1
!----- species(1) % N_of_atoms    = 68     ! <== # of atoms in a molecule of species 1
!----- species(1) % flex         = true     ! <== Flexible : true , false
!----- species(2) % residue       = CCC      ! <== Residue label for species 2
!----- species(2) % N_of_molecules = 1      ! <== # of molecules of species 2
!----- species(2) % N_of_atoms    = 383    ! <== # of atoms in a molecule of species 2
!----- species(2) % flex         = false
!
!----- AD-HOC settings
!----- OPT_parms = true
!----- ad_hoc   = true
!----- ad_hoc:QM:residue(67:68)=CCC
!----- ad_hoc:QM:nr(67:68)=2
!
!----- ENVIRONMENT parameters ...
!----- thermostat        = Microcanonical ! <== Berendsen, Nose_Hoover, Microcanonical
!----- cutoff_radius      = 50.0d0          ! <== Cut off radius (Angs.) for electrostatic
!----- damping_Wolf       = 0.001           ! <== damping parameter (Angs.^-1)
!----- driver_MM          = MM_Dynamics    ! <== MM_Dynamics , MM_Optimize , NormalModes , Parametrize
!----- read_velocities    = true            ! <== reads the initial velocities : T_ , F_
!----- MM_input_format    = GMX             ! <== GMX, NAMD, GAFF

```

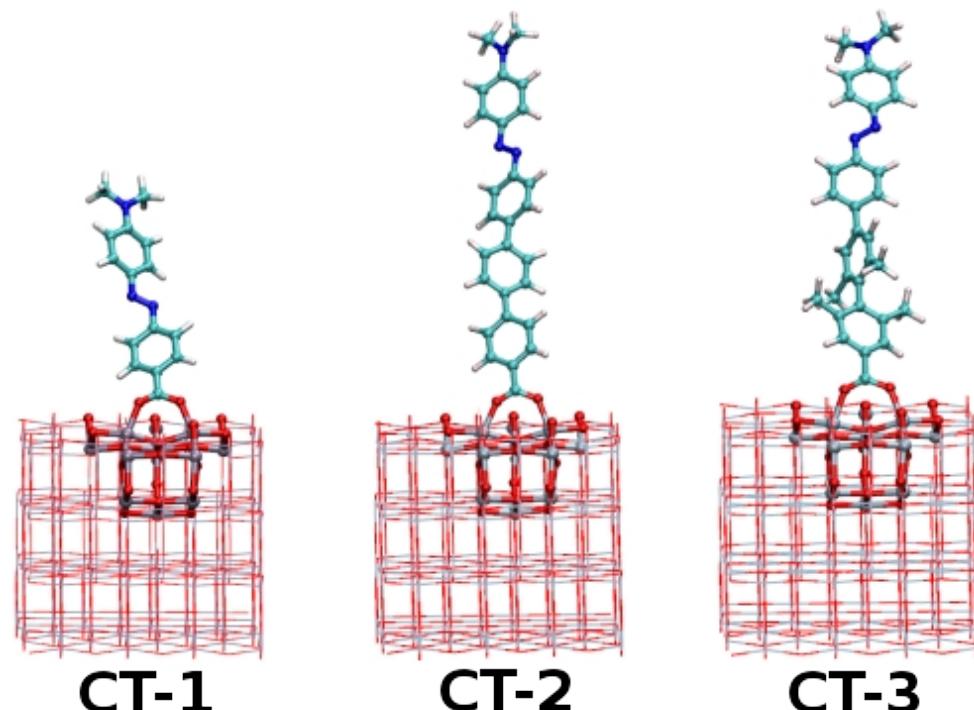
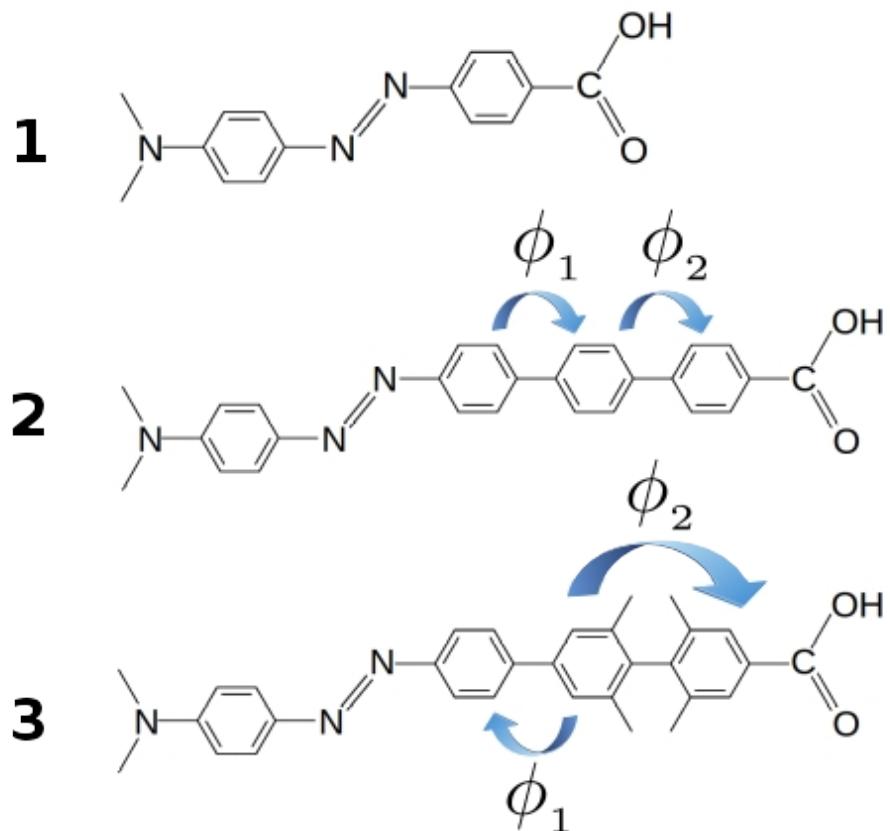


# Dependence of single-molecule junction conductance on molecular conformation

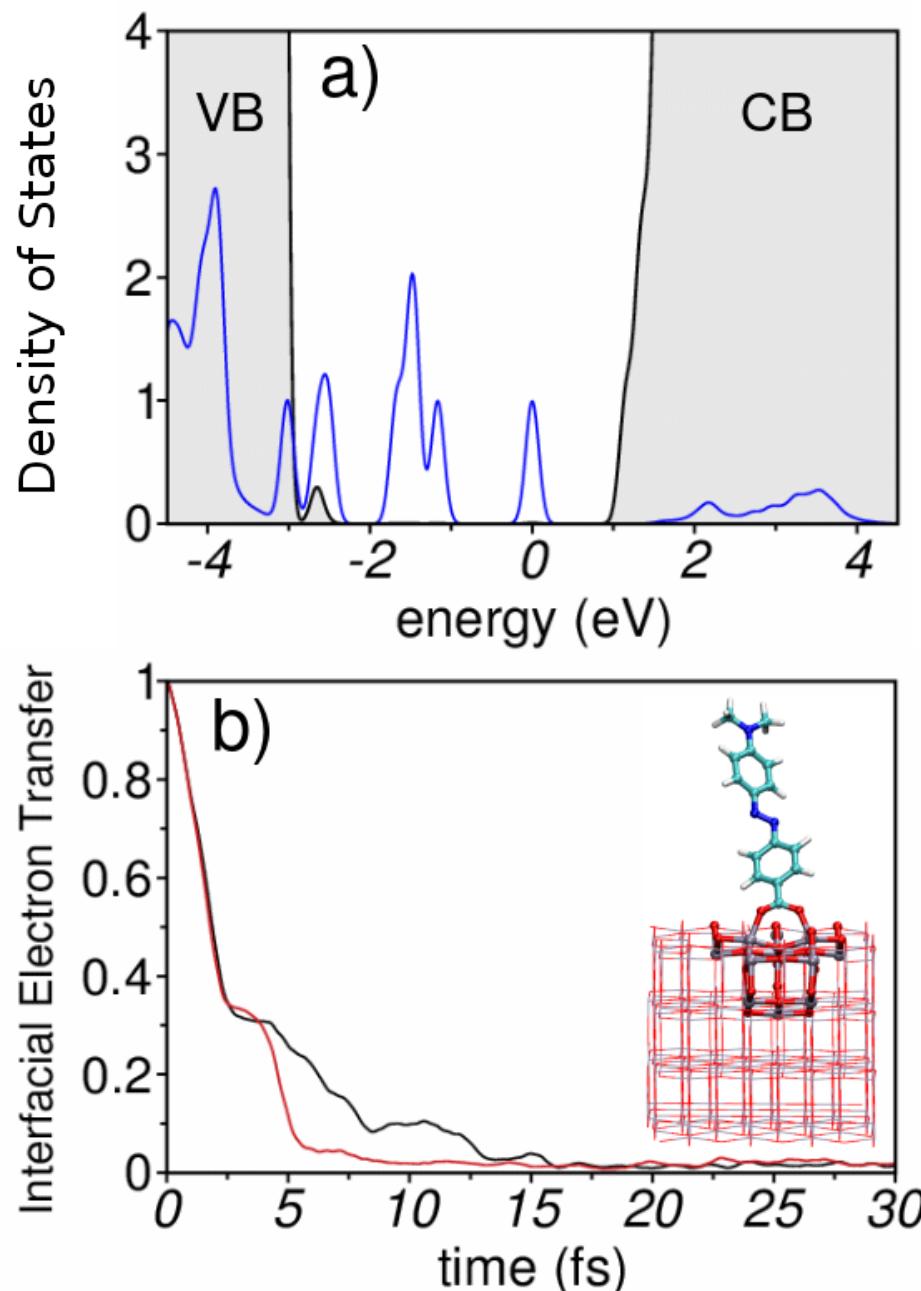
Latha Venkataraman<sup>1,4</sup>, Jennifer E. Klare<sup>2,4</sup>, Colin Nuckolls<sup>2,4</sup>, Mark S. Hybertsen<sup>3,4</sup> & Michael L. Steigerwald<sup>2</sup>



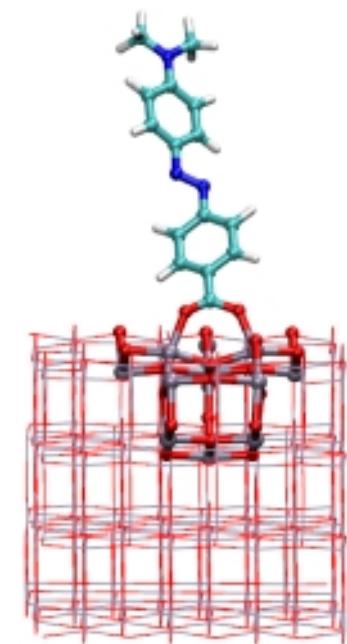
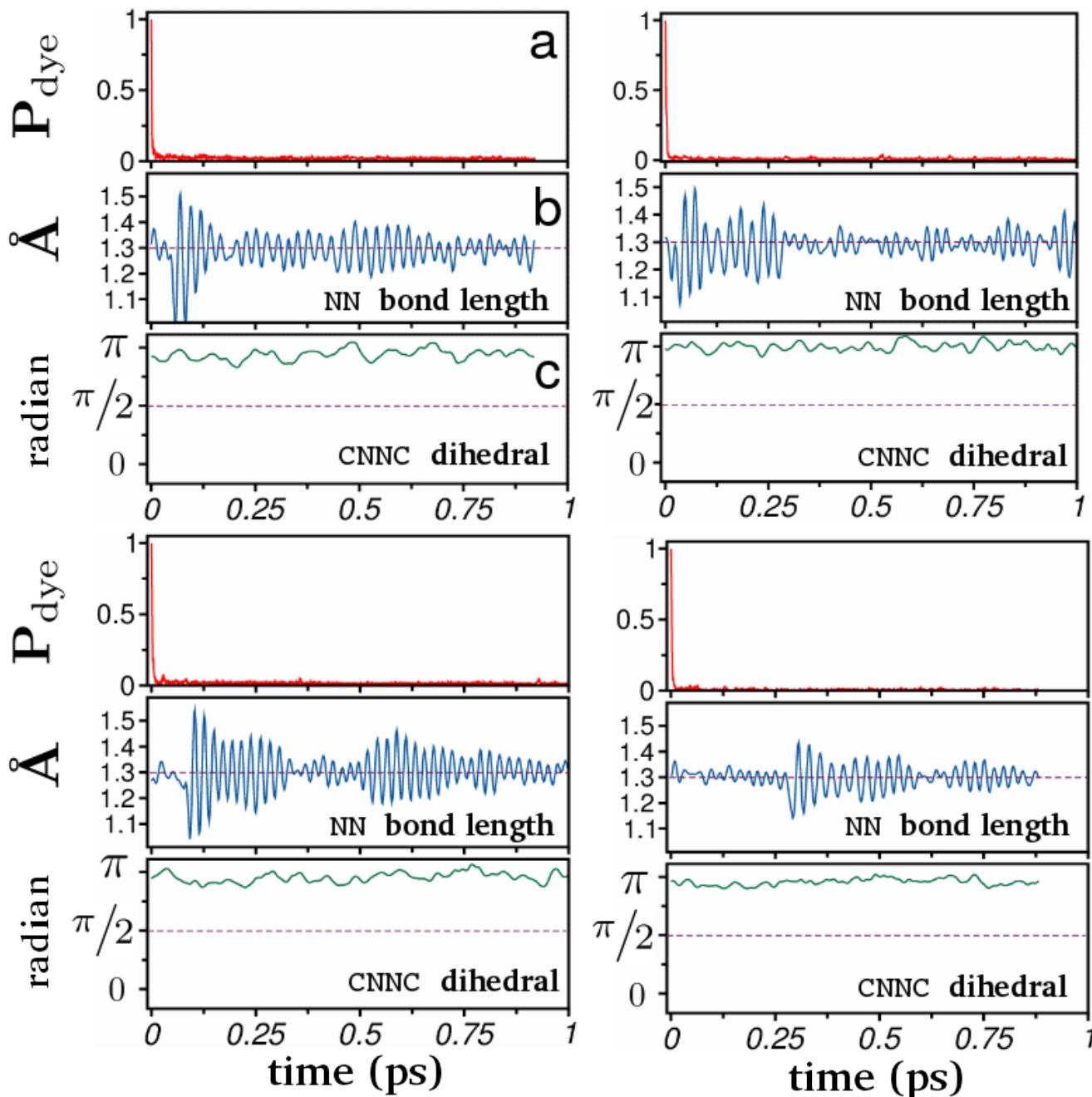
# Charge Transfer *vs* Structural Relaxation



# Charge Transfer *vs* Structural Relaxation

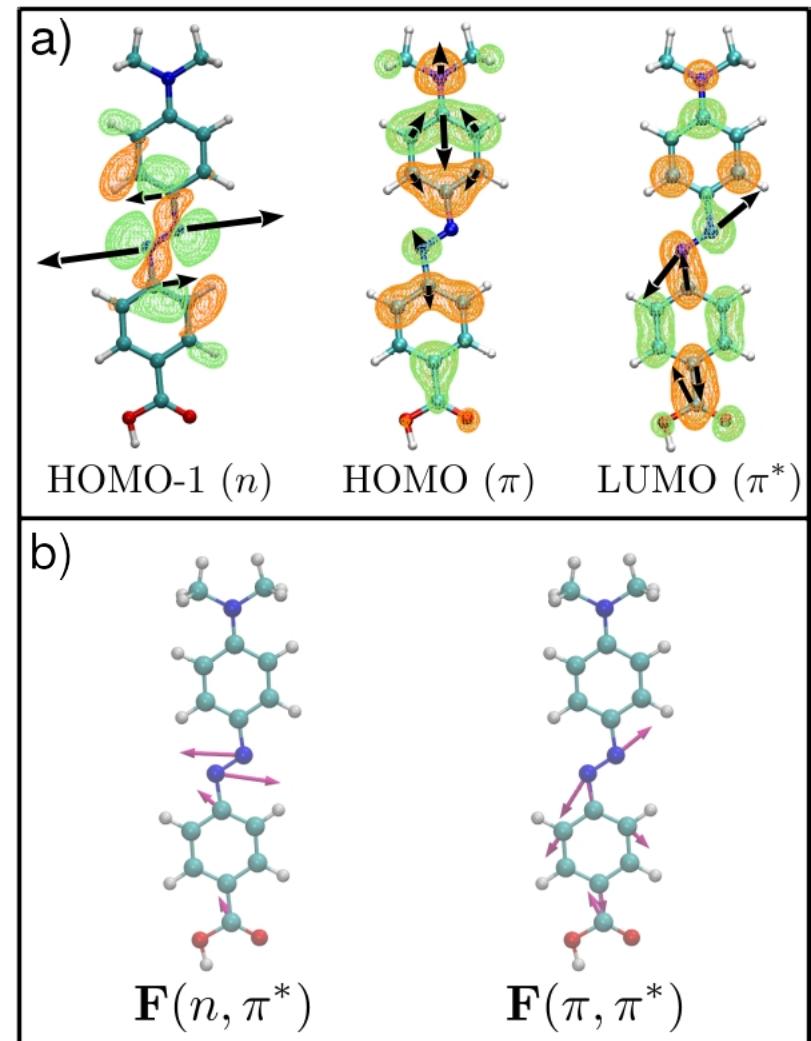
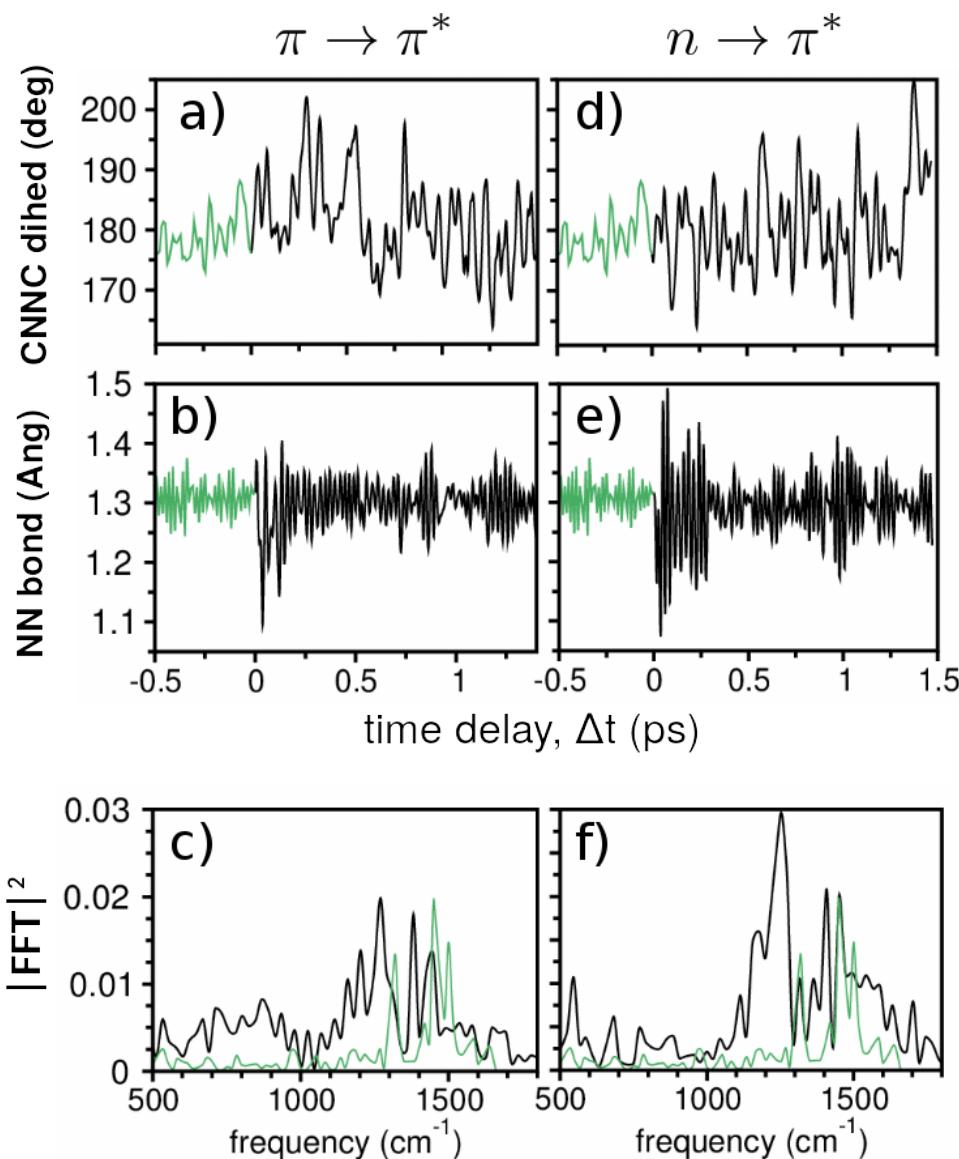


# Charge Transfer *vs* Structural Relaxation

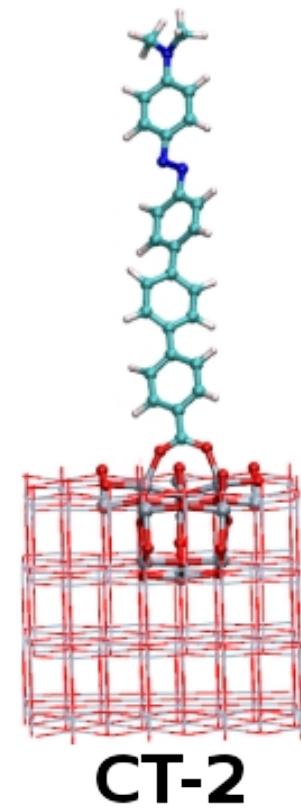
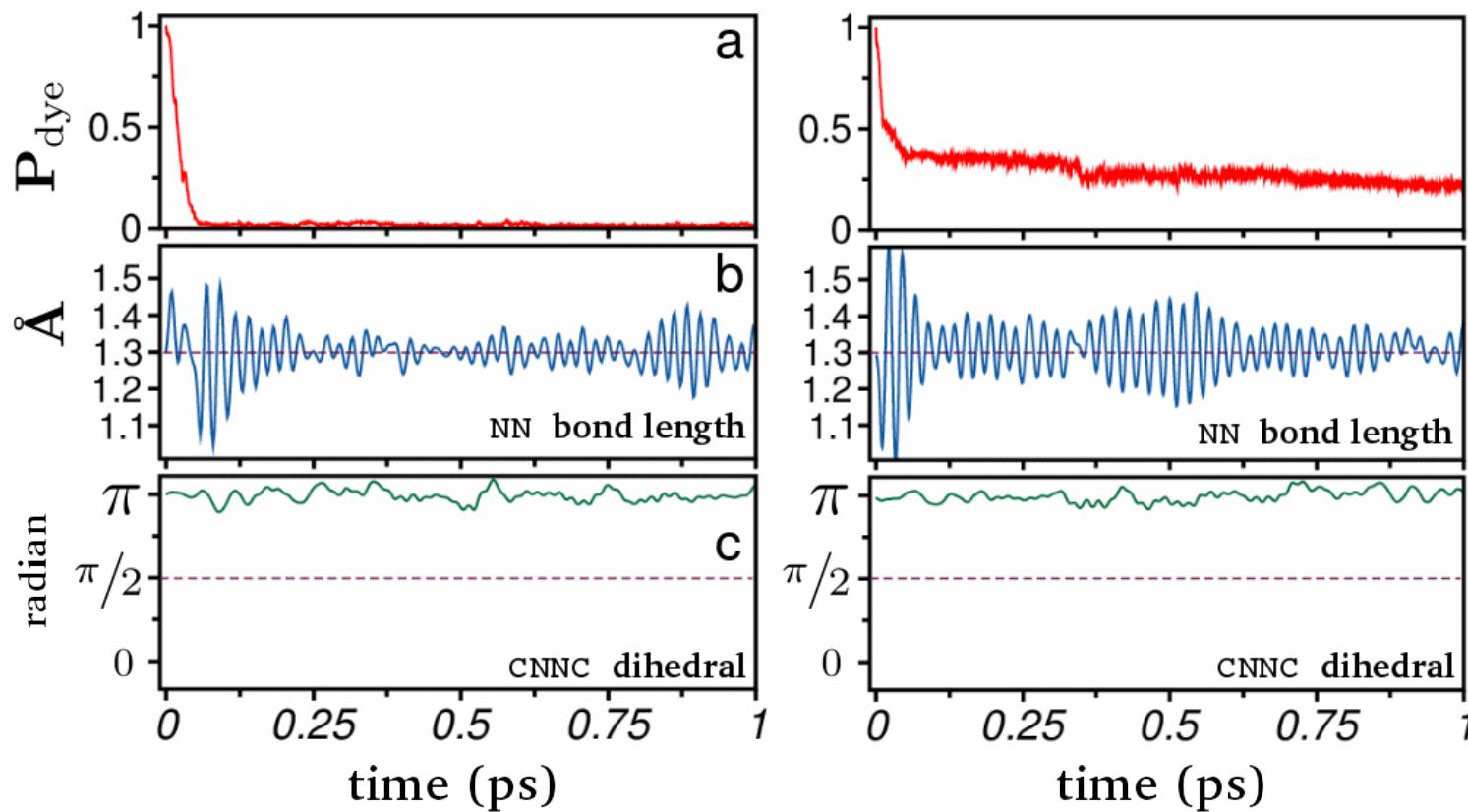


**CT-1**

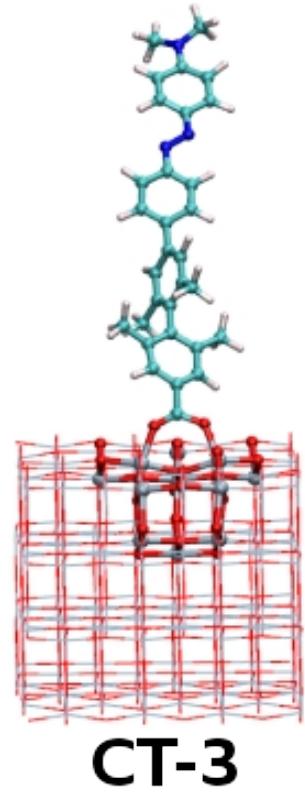
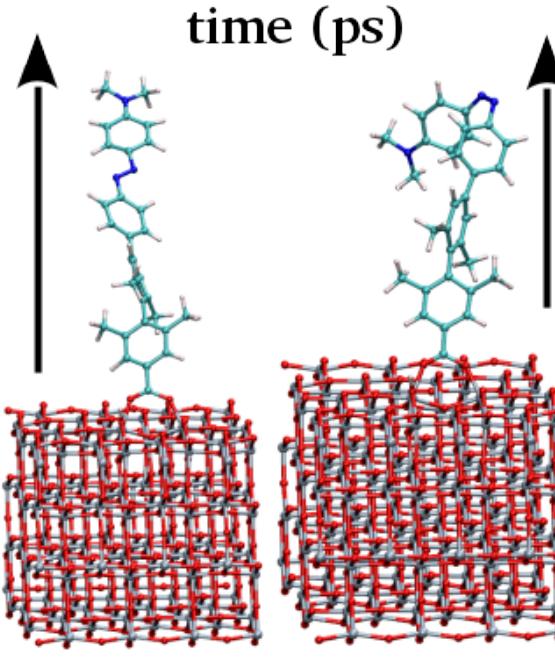
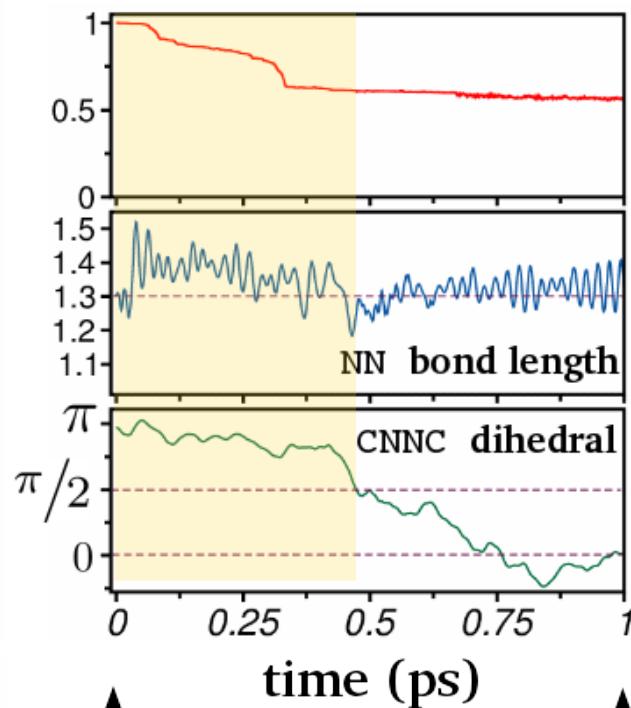
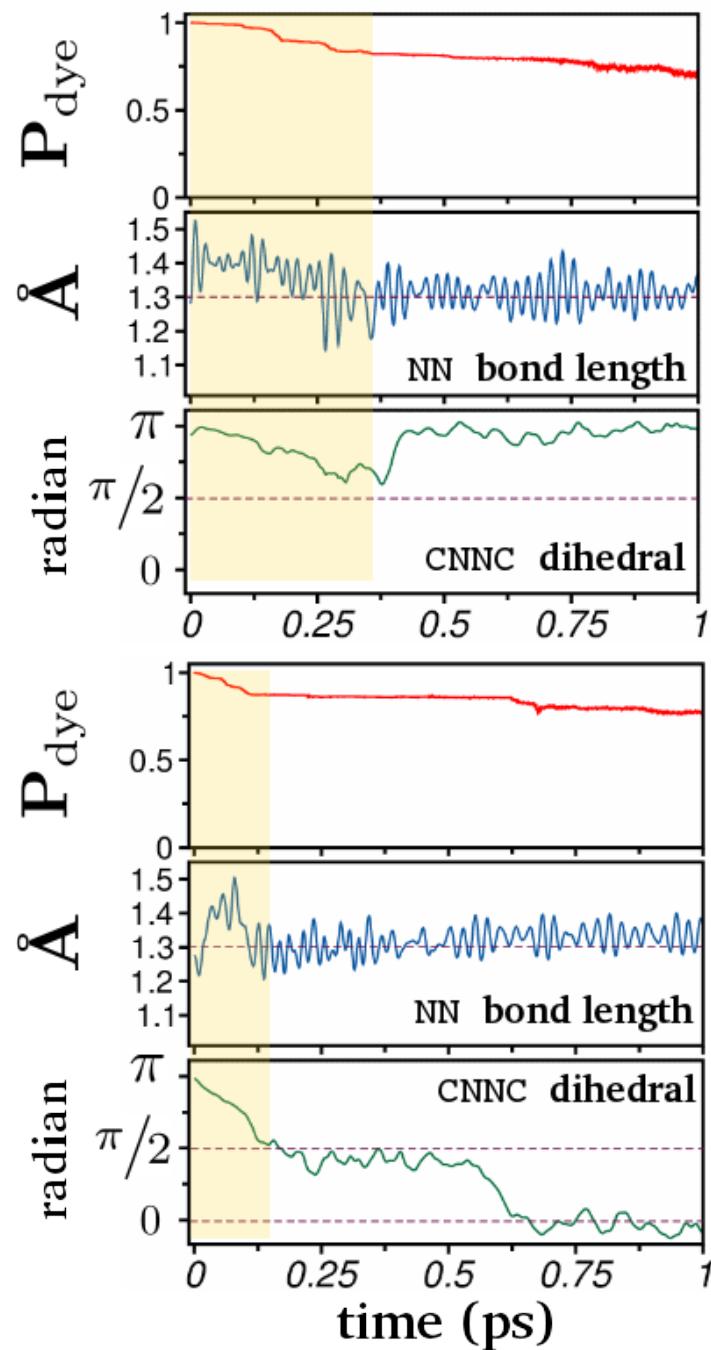
# Charge Transfer *vs* Structural Relaxation



# Charge Transfer *vs* Structural Relaxation

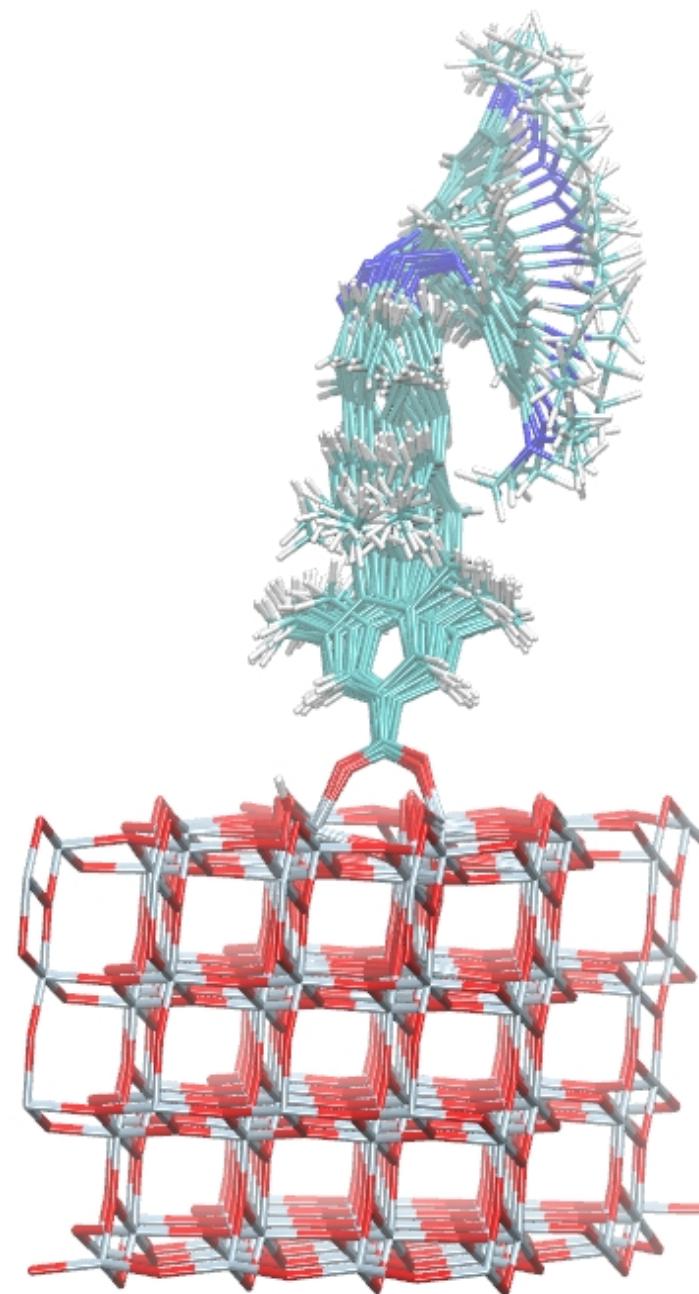
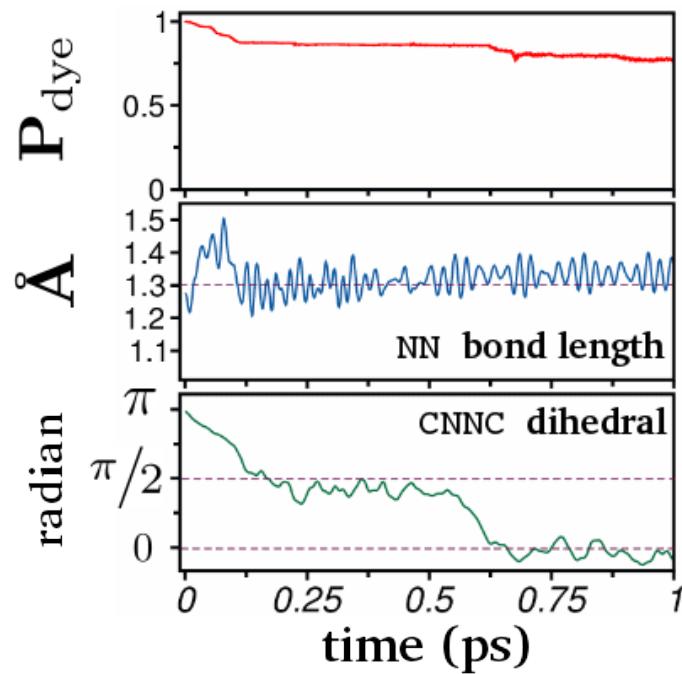


# Charge Transfer *vs* Structural Relaxation



**CT-3**

# Charge Transfer *vs* Structural Relaxation





# MAPbI<sub>3</sub> Hybrid Perovskite: Setting Up the System in card.inpt

Example: Classical MD of 6x6x6 MAPbI<sub>3</sub> Hybrid Perovskite

card.inpt

```
! ACTION flags

DRIVER      = MM_Dynamics

nuclear_matter = MDynamics
file_type      = structure
file_format    = pdb
                                         ! <== structure or trajectory
                                         ! <== xyz , pdb or vasp

PBC = [ 1 , 1 , 1 ]                                ! Periodic Boundary Conditions on {x,y,z} directions

t_f  = 10.d00
n_t  = 20000
                                         ! <== final time in PIC0oseconds

VDOS = true
                                         ! velocity-DOS is calculated on the fly

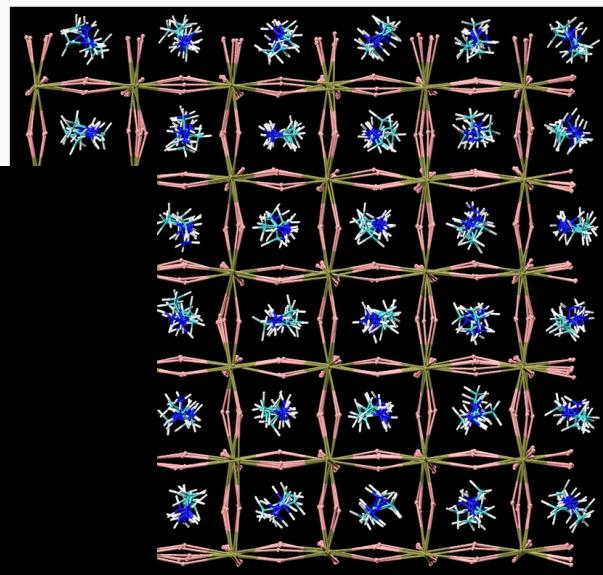
! SYSTEM INFO

N_of_molecules = 217
N_of_species   = 2
                                         ! <== total number of molecules
                                         ! <== total number of species

! repeat the following information filling for all the different species ...

species(1) % residue      = MA+       ! <== Residue label for species 1
species(1) % N_of_molecules = 216
species(1) % N_of_atoms    = 8        ! <== Number of molecules of species 1
                                         ! <== Number of atoms comprising a single molecule of species 1
species(1) % flex          = true
                                         ! <== Flexible : true , false

species(2) % residue      = PbI        ! <== Residue label for species 2
species(2) % N_of_molecules = 1
species(2) % N_of_atoms    = 864
species(2) % flex          = true
```



Files:

- input.pdb
- MA+.psf
- PbI.psf
- input.prm

# MAPbI<sub>3</sub> Hybrid Perovskite: Setting Up the System in card.inpt

Example: Classical MD of 6x6x6 MAPbI<sub>3</sub> Hybrid Perovskite

MA+.psf

```
REMARKS methylammonium cation (CH3NH3)+

8 !NATOM
 1 SYS      1       MA+      H       HP      0.540000    1.0080
 2 SYS      1       MA+      H       HP      0.540000    1.0080
 3 SYS      1       MA+      H       HP      0.540000    1.0080
 4 SYS      1       MA+      N       N3     -1.100000   14.011
 5 SYS      1       MA+      C       CT      0.771000   12.011
 6 SYS      1       MA+      H       HC      0.023000    1.0080
 7 SYS      1       MA+      H       HC      0.023000    1.0080
 8 SYS      1       MA+      H       HC      0.023000    1.0080

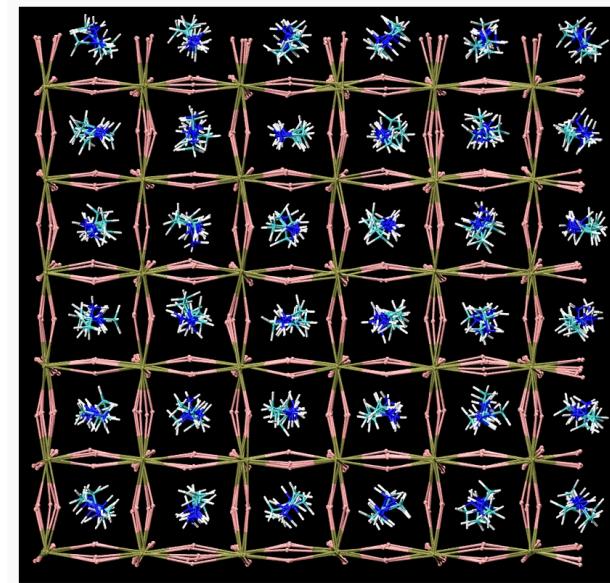
7 !NBOND: bonds
 5   6           5   7           5   8           4   1
 4   2           4   3           4   5

12 !NTHETA: angles
 7   5   8           6   5   7           6   5   8
 3   4   5           2   4   3           2   4   5
 1   4   2           1   4   3           1   4   5
 4   5   6           4   5   7           4   5   8

9 !NPHI: dihedrals
 3   4   5   6           3   4   5   7
 3   4   5   8           2   4   5   6
 2   4   5   7           2   4   5   8
 1   4   5   7           1   4   5   8
 1   4   5   6

0 !NIMPHI: impropers

0 !AD-HOC: flex
```



Files:

- input.pdb
- MA+.psf
- Pbl.psf
- input.prm

# MAPbI<sub>3</sub> Hybrid Perovskite: Setting Up the System in card.inpt

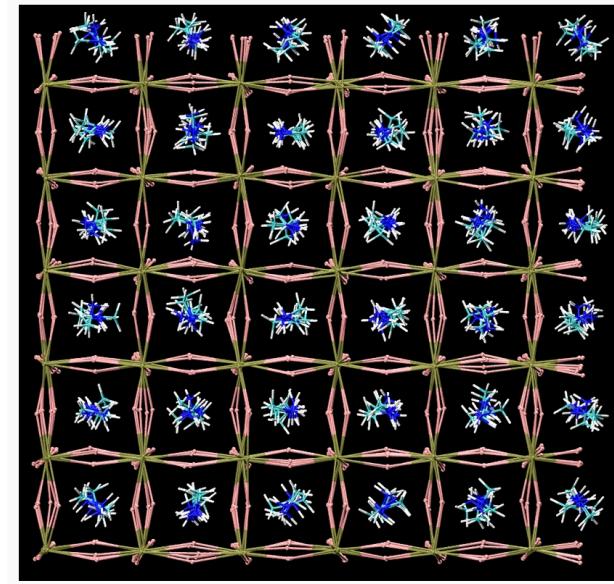
Example: Classical MD of 6x6x6 MAPbI<sub>3</sub> Hybrid Perovskite

PbI.psf

```
REMARKS PbI cage

 864!NATOM
 1  SYS   1      PBI    Pb     Pb      2.0300    207.20
 2  SYS   1      PBI    Pb     Pb      2.0300    207.20
 3  SYS   1      PBI    Pb     Pb      2.0300    207.20
 4  SYS   1      PBI    I      I      -1.1300   126.90
 5  SYS   1      PBI    I      I      -1.1300   126.90
 .
 .
 .
 .
 .
 .
 .
 .
 .
 850 SYS   1      PBI    I      I      -1.1300   126.90
 851 SYS   1      PBI    I      I      -1.1300   126.90
 852 SYS   1      PBI    I      I      -1.1300   126.90
 853 SYS   1      PBI    Pb     Pb      2.0300    207.20
 854 SYS   1      PBI    Pb     Pb      2.0300    207.20
 855 SYS   1      PBI    Pb     Pb      2.0300    207.20
 856 SYS   1      PBI    I      I      -1.1300   126.90
 857 SYS   1      PBI    I      I      -1.1300   126.90
 858 SYS   1      PBI    I      I      -1.1300   126.90
 859 SYS   1      PBI    I      I      -1.1300   126.90
 860 SYS   1      PBI    I      I      -1.1300   126.90
 861 SYS   1      PBI    I      I      -1.1300   126.90
 862 SYS   1      PBI    I      I      -1.1300   126.90
 863 SYS   1      PBI    I      I      -1.1300   126.90
 864 SYS   1      PBI    I      I      -1.1300   126.90

 0 !NBOND: bonds
 0 !NTHETA: angles
 0 !NPHI: dihedrals
 0 !NIMPHI: impropers
 0 !AD-HOC: flex
```



Files:

- input.pdb
- MA+.psf
- PbI.psf
- input.prm

# MAPbI<sub>3</sub> Hybrid Perovskite: Setting Up the System in card.inpt

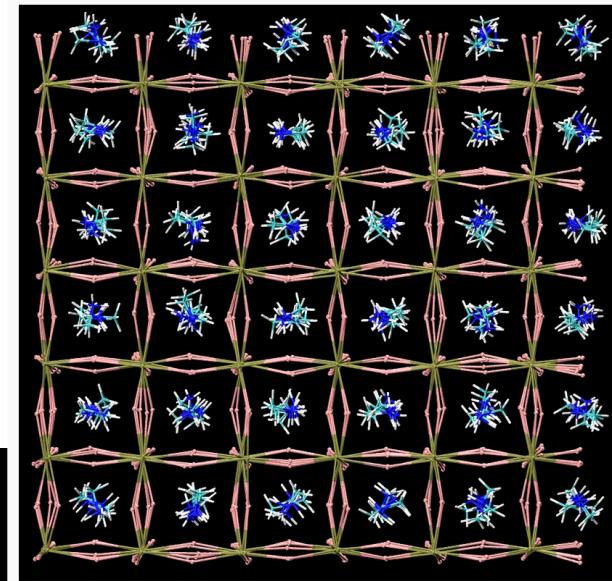
Example: Classical MD of 6x6x6  
MAPbI<sub>3</sub> Hybrid Perovskite

card.inpt, continued.

already thermalized

Files:

- input.pdb
- MA+.psf
- PbI.psf
- input prm



```
!-----  
! ENVIRONMENT parameters ...  
  
thermostat      = Berendsen          ! <== Berendsen, Nose_Hoover, Microcanonical  
temperature       = 300.d0            ! <== Bath Temperature (K)  
thermal_relaxation_time = 3.d00        ! <== Temperature coupling term with the bath  
                                      ! <== SMALL = STRONG coupling ; use "infty" to decouple  
cutoff_radius     = 25.d0             ! <== Cut off radius (Angs.) for electrostatic and LJ interactions  
damping_Wolf      = 0.005             ! <== Wolf's method damping parameter (length^{-1});  
                                      ! (J. Chem. Phys. 1999; 110(17):8254)  
!  
! GENERAL INFO ...  
!  
driver_MM         = "MM_Dynamics"      ! <== MM_Dynamics , MM_Optimize , NormalModes , Parametrize  
read_velocities   = true              ! <== reads the initial velocities : T_ , F_  
MM_input_format   = GAFF              ! <== GMX, NAMD, GAFF  
MM_log_step        = 1                 ! <== step for saving MM results & parameters  
MM_frame_step      = 25                ! <== step for saving MM results & parameters
```

# MAPbI<sub>3</sub> Hybrid Perovskite: Setting Up the System in card.inpt

```
*>>> CHARMM Parameter file <<<
FF-sets
! comb-rule      SCNB      SCEE
    2           2.0       1.2

ATOMS
MASS   12 CT     12.01000
MASS   20 HC     1.00800
MASS   24 HP     1.00800
MASS   00 N3     14.01000

BONDS
!
!V(bond) = Kb(b - b0)**2
!Kb: kcal/mole/A**2
!b0: A
!
!atom type    Kb      b0
CT    HC     338.70  1.0910  *
CT    N3     293.60  1.4990  *
N3    HP     369.00  1.0330  *

ANGLES
!
!V(angle) = Ktheta(Theta - Theta0)**2
!Ktheta: kcal/mole/rad**2
!Theta0: degrees
!Kub: kcal/mole/A**2 (Urey-Bradley)
!
!atom types      Ktheta    Theta0    Kub    S0
HC    CT    HC     39.00  110.74  *
HP    N3    CT     46.20  110.11  *
HP    N3    HP     40.50  108.11  *
N3    CT    HC     49.00  107.91  *

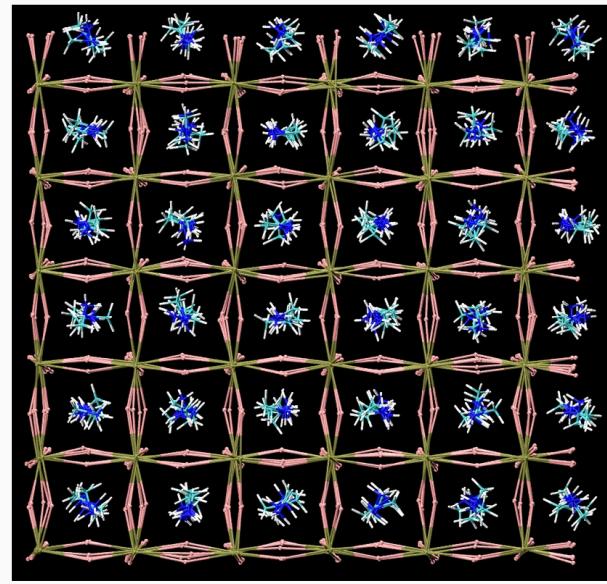
DIHEDRALS
!
!V(dihedral) = Kchi(1 + cos(n(chi) - delta))
!Kchi: kcal/mole
!n: multiplicity
!delta: degrees
!
!atom types          Kchi    n    phase
X    N3    CT    X     0.1556  3    0.00  *

IMPROPERs
!
!V(improper) = Kpsi(psi - psi0)**2
!Kpsi: kcal/mole/rad**2
!psi0: degrees
!
!atom types          Kpsi          psi0
!
```

⇐ input.prm

Intra-molecular  
Bonding FF parameters

Relevant entries highlighted



Example: Classical MD of 6x6x6 MAPbI<sub>3</sub> Hybrid Perovskite

Files:

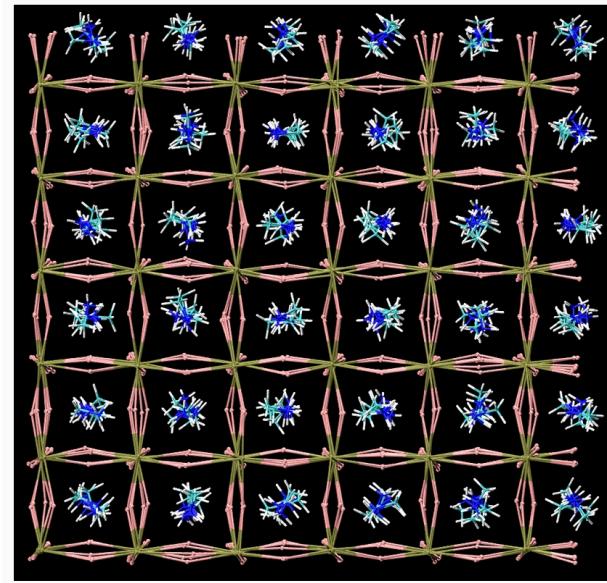
- input.pdb
- MA+.psf
- Pbl.psf
- input.prm

# MAPbI<sub>3</sub> Hybrid Perovskite: Setting Up the System in card.inpt

## NONBONDED

```
!
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!epsilon: kcal/mole, Eps,i,j = sqrt(eps,i * eps,j)
!Rmin/2: A, Rmin,i,j = Rmin/2,i + Rmin/2,j
!
!atom    type      epsilon      Rmin/2      eps,1-4      Rmin/2,1-4
CT      LJ       -0.109400     1.908000    -0.054700     1.908000
HC      LJ       -0.015700     1.100000    -0.007850     1.100000
N3      LJ       -0.170000     1.824000    -0.085000     1.842000
HP      LJ       -0.015700     0.600000    -0.007850     0.600000
!
!V(Buckingham) = A*exp(-rij/B) - C/rij**6
!A: kcal/mole, Aij = sqrt(Ai * Aj)
!B^-1: Angstrom, 1/Bij = (1/Bi + 1/Bj)/2
!C: kcal/mole, Cij = sqrt(Ci * Cj)
!
!atom    type      A          B          C
Pb      buck   70359906.63  0.131258  0.0000
I       buck   22793.3385   0.482217  696.949542
```

⇐ input.prm  
(continued)



## SPECIALPAIRS

```
!
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!epsilon: kcal/mole, Eps,i,j = sqrt(eps,i * eps,j)
!Rmin/2: A, Rmin,i,j = Rmin/2,i + Rmin/2,j
!
!atom1 atom2    type      epsilon      Rmin/2      eps,1-4      Rmin/2,1-4
Pb      HP      LJ       -0.014000     1.27093    -0.007000     1.27093
Pb      HC      LJ       -0.014000     1.52088    -0.007000     1.52088
I       HP      LJ       -0.057400     1.54338    -0.028700     1.54338
I       HC      LJ       -0.057400     1.73982    -0.028700     1.73982
!
!V(Buckingham) = A*exp(-rij/B) - C/rij**6
!A: kcal/mole, Aij = sqrt(Ai * Aj)
!B: Angstrom, 1/Bij = (1/Bi + 1/Bj)/2
!C: kcal/mole, Cij = sqrt(Ci * Cj)
!
!atom1 atmo2    type      A          B          C
Pb      Pb      buck   70359906.63  0.131258  0.0000
I       I       buck   22793.3385   0.482217  696.949542
Pb      I       buck   103496.1330   0.321737  0.000000
Pb      N3     buck   32690390.93   0.150947  0.000000
Pb      CT      buck   32690390.93   0.150947  0.000000
I       N3     buck   112936.7142   0.342426  0.000000
I       CT      buck   112936.7142   0.342426  0.000000
```

Non-Bonding FF parameters

Relevant entries highlighted

Files:

- input.pdb
- MA+.psf
- Pbl.psf
- input.prm

# Density of Vibrational states (VDOS)

velocity power spectrum (**VelPS.dat**) for stationary state:

$$\sum_i^N |\mathbf{v}_i(\omega)|^2 = \sum_i^N \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} \mathbf{v}_i(t + \tau) \cdot \mathbf{v}_i(\tau) d\tau \right) e^{i\omega t} dt$$

$$\text{VelPS}(\omega) \equiv \sum_i^N |\mathbf{v}_i(\omega)|^2 = \sum_i^N \int_{-\infty}^{\infty} \langle \mathbf{v}_i(\tau + t) \cdot \mathbf{v}_i(t) \rangle_{\tau} e^{i\omega t} dt$$

Vibrational Density of States (**VDOS.dat**):

$$VACF_{mw}(t) = \frac{\sum_i m_i \langle \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \rangle}{\sum_i m_i \langle \mathbf{v}_i(0) \cdot \mathbf{v}_i(0) \rangle} = \frac{\sum_i m_i \langle \mathbf{v}_i(t) \cdot \mathbf{v}_i(0) \rangle}{3Nk_B T}$$

$$\dot{\mathbf{r}}_k(t) = \mathbf{v}_k = \sum_s^{modes} \mathbf{Q}_{sk}(-i\omega_s) e^{-i\omega_s t} \quad \xrightarrow{\text{thermal equil. + eqiupartition}} \quad m_k |\mathbf{Q}_{sk}|^2 \omega_s^2 = 3k_B T$$

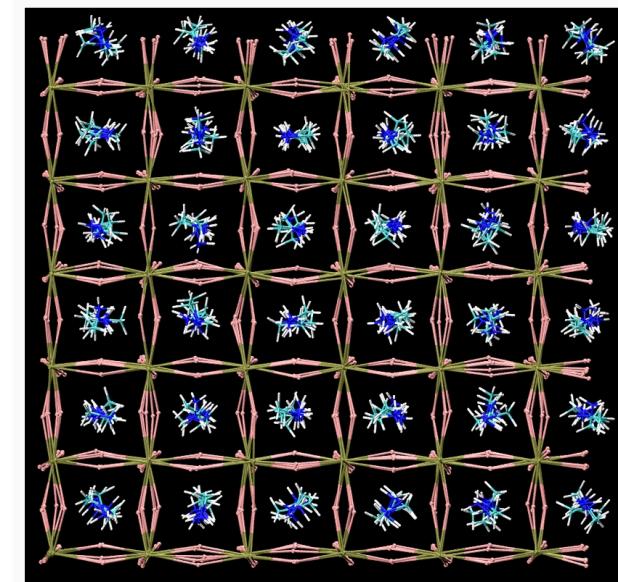
$$\sum_i^N m_i |\mathbf{v}_i(\omega)|^2 = \sum_s \sum_{k=1}^N \int_{-\infty}^{\infty} m_k |\mathbf{Q}_{sk}|^2 \omega_s^2 e^{i(\omega + \omega_s)t} dt = 3Nk_B T \sum_s \delta(\omega + \omega_s)$$

$$\begin{aligned} \text{VDOS}(\omega) \equiv \rho(\omega) &= \sum_s \delta(\omega + \omega_s) \\ &= \frac{\sum_i^N m_i |\mathbf{v}_i(\omega)|^2}{3Nk_B T} = \frac{1}{3Nk_B T} \int_0^{\infty} \sum_i^N m_i \langle \mathbf{v}_i(\tau + t) \cdot \mathbf{v}_i(t) \rangle_{\tau} e^{i\omega t} dt \end{aligned}$$

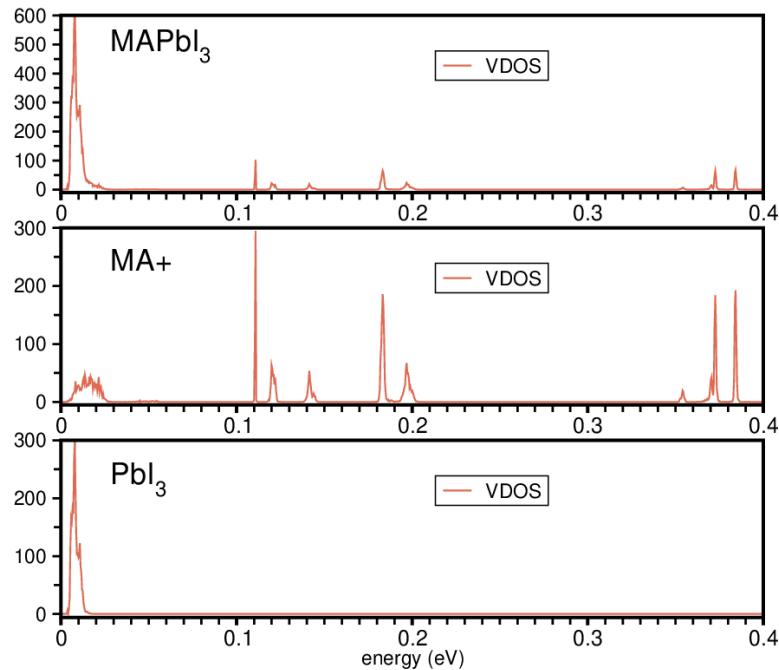
# Density of Vibrational states (VDOS)

card.inpt

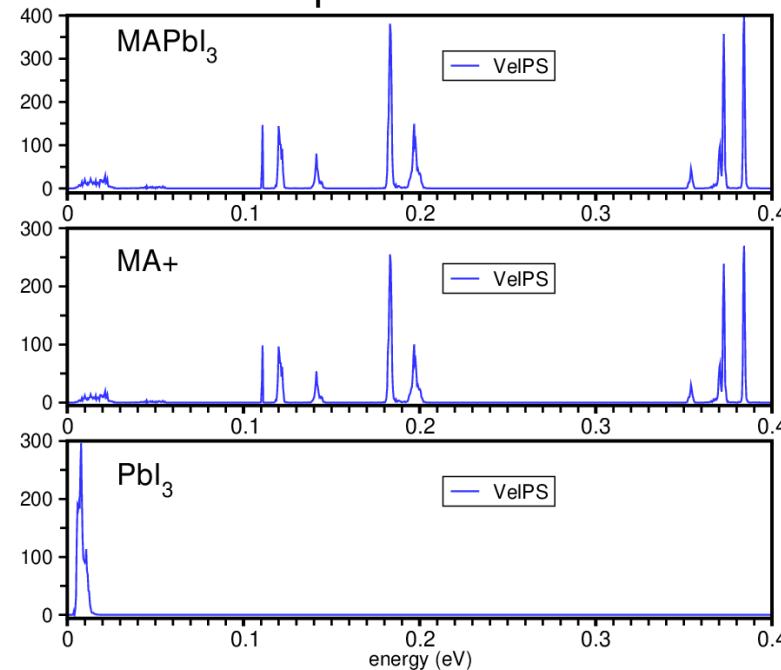
```
!-----  
! ACTION      flags  
  
DRIVER        = MM_Dynamics  
  
nuclear_matter = MDynamics  
file_type      = structure  
file_format    = pdb  
  
PBC = [ 1 , 1 , 1 ]  
          ! <= structure or trajectory  
          ! <= xyz , pdb or vasp  
  
t_f  = 10.d00  
          ! Periodic Boundary Conditions on {x,y,z} directions  
n_t  = 20000  
          ! <= final time in PIC0oseconds  
  
VDOS = true  
          ! velocity-DOS is calculated on the fly
```



Vibrational DoS



Power Spectrum of velocities



# Running Dynemol

Setup of environment variables:

```
export DYNEMOLDIR= "path to dynemol executable directory"
```

```
export DYNEMOLWORKDIR= $(pwd)
```

Must have in work-directory \$DYNEMOLWORKDIR:

- the appropriate input files for the job
- file with execution directions: *card.inpt*
- Execution script (e.g., run-SN.sh, for Single-Node execution)

# Running Dynemol

In directory \$DYNEMOLWORKDIR:

- the execution script run-SN.sh (basic model of script)

```
#!/bin/bash

#####
#Script Name    : run-SN.sh
#Description   : execute Dynemol from pwd using local dynemol src files
#Args          : none
#Author        : Luis G C Rego
#####
#!/bin/bash

export DYNEMOLWORKDIR=$(pwd)
export DYNEMOLDIR=/home/lrego/QMMM/development/dynemol-dir

$DYNEMOLDIR/dynemol
```

- On the terminal run  
> ./run-SN.sh

# Running Dynemol

In directory \$DYNEMOLWORKDIR:

- Outputs , results , log-files , security-files , etc. are stored in:
  - \$DYNEMOLWORKDIR/ancillary.trunk/
  - \$DYNEMOLWORKDIR/dos.trunk/
  - \$DYNEMOLWORKDIR/dyn.trunk/
  - \$DYNEMOLWORKDIR/log.trunk/
  - \$DYNEMOLWORKDIR/MO.trunk/
  - \$DYNEMOLWORKDIR/opt.trunk/
- *which are deleted and made anew at every execution.*
- *each directory gets a copy of card.inpt for future reference.*

# Acknowledgements

## Have contributed to the DynEMol Project:

- Robson O. Silva
- Alberto Torres
- Diego Hoff
- Grazielle Bertolini
- João V. Meyer

