

Introduction to QXMD

Subodh Tiwari

Lindsay Bassman, Aravind Krishnamoorthy

*Collaboratory for advanced computing and Simulation
Department of Material Science & Department of Physics
University of Southern California*



MAGICS Material Software Workshop

Los Angeles

Nov 12-15, 2017



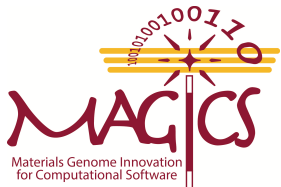
Acknowledgement

“This work was supported as part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award Number *DE-SC00014607*.”



Capabilities

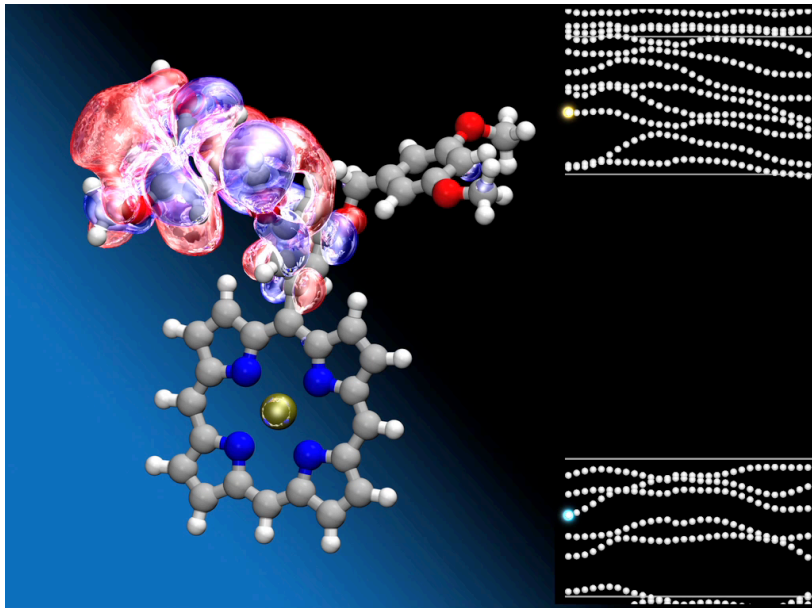
QXMD is scalable parallel quantum molecular dynamics engine.



Capabilities

QXMD is scalable parallel quantum molecular dynamics engine.

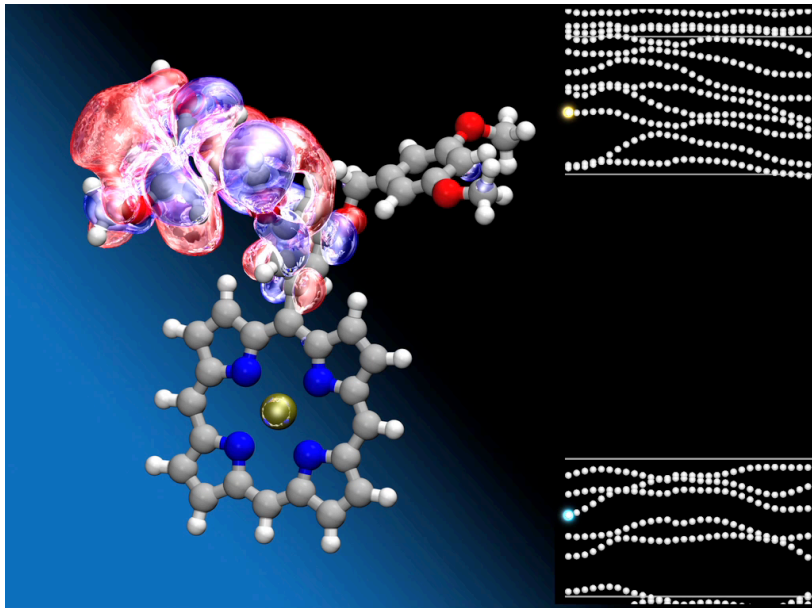
Non-adiabatic Quantum Molecular Dynamics (NAQMD)



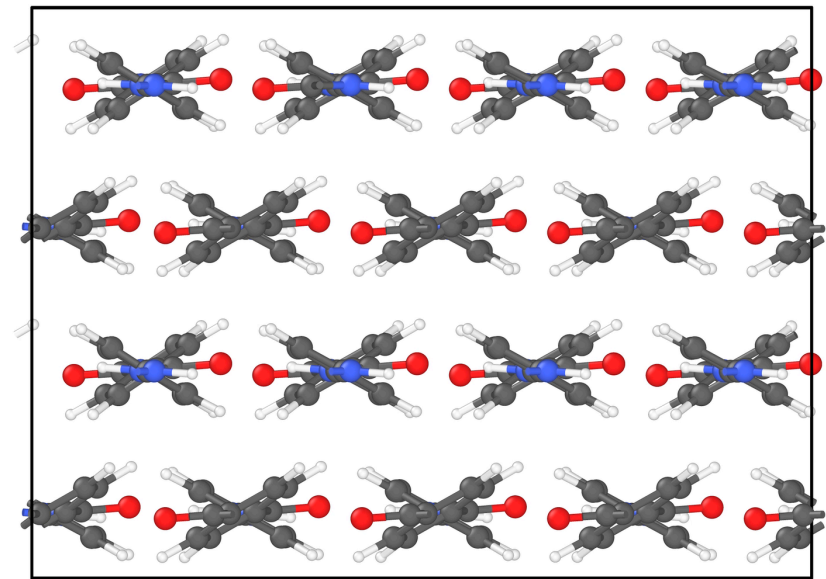
Capabilities

QXMD is scalable parallel quantum molecular dynamics engine.

Non-adiabatic Quantum Molecular Dynamics (NAQMD)

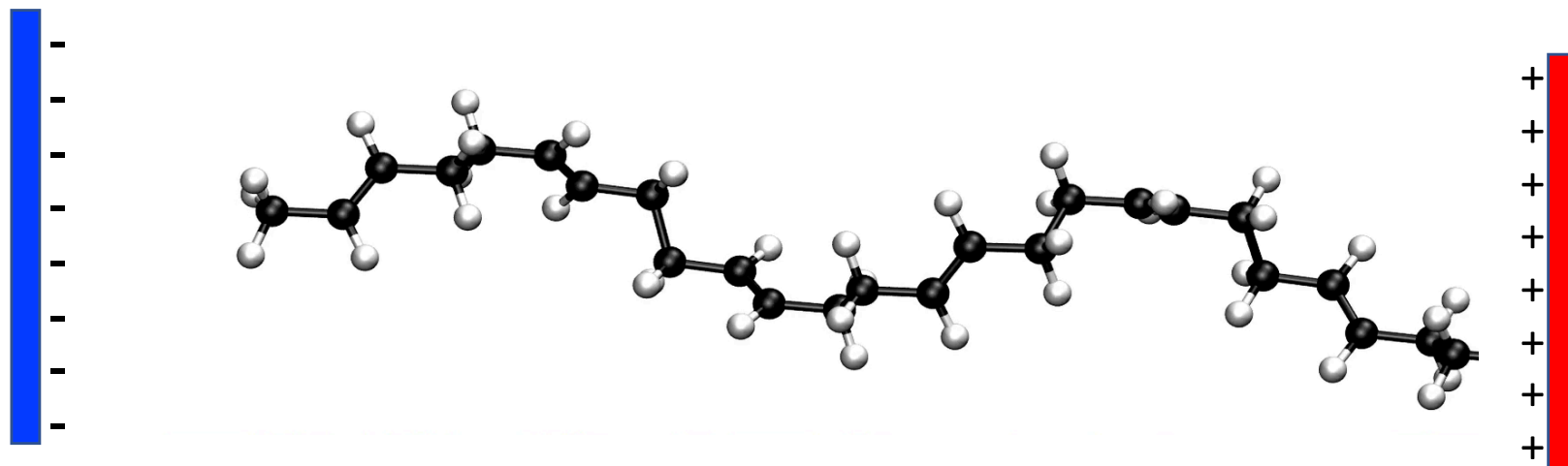


Multiscale Shock theory (MSST)



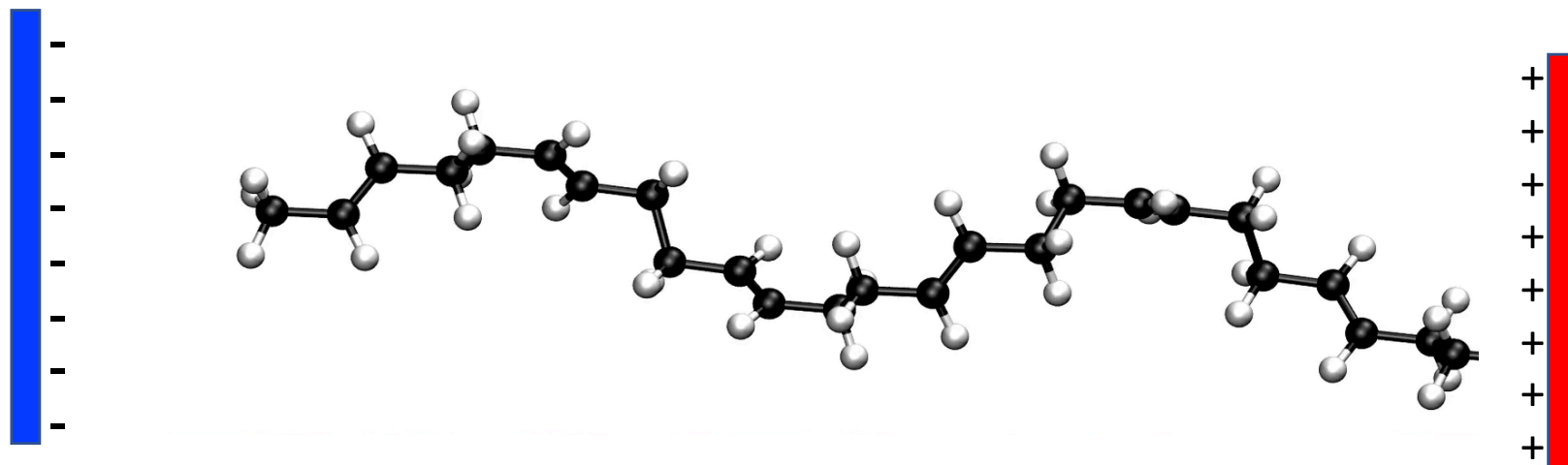
Capabilities

External Electric Field

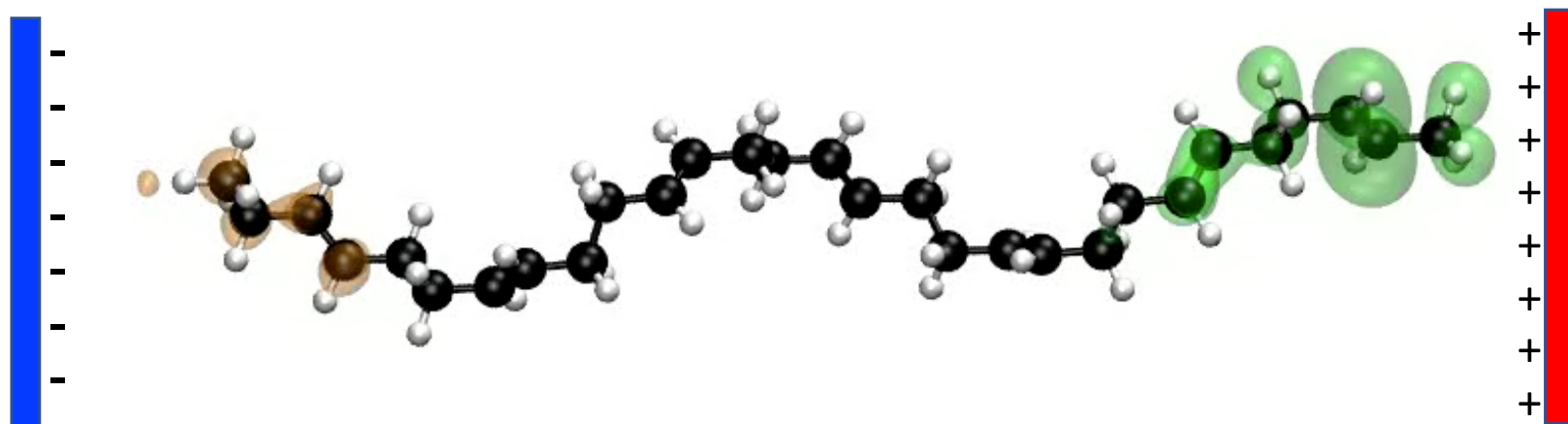


Capabilities

External Electric Field



NAQMD under External Electric Field



Outline

1- Optimization of Geometry

- Hands-on I: Optimization of water
- Hands-on II: Optimization of MoSe_2

2- Non-adiabatic Quantum Molecular Dynamics

- Hands-on III: Excited state dynamics of MoSe_2

Download

Download from **MAGICS** website

<https://magics.usc.edu/home-old/software-downloads/>

Software Download Links

EXECUTABLE



Cray XC_40

ALCF Theta, Cray Intel

Compiler



IBM BG/Q

ALCF Mira, mpiwrapper-xl



USC-HPC

CentOS 7.3, ifort 16.0,

OpenMPI 1.8.8

Download on your computer and use file transfer tool (**Filezilla**, **Fetch** or **scp**) to move software from your computer to USC-HPC.

Software Package

Untar file using command

```
$tar -xvf PWP-CentOS7.3.1611Core-ifort16.0.0-OpenMPI1.8.8.tar
```

Each software contains program, example, README

```
$cd program
```

Contains different example

Contains executable, input template file

Readme file

```
.
├── examples
├── QXMD_Manual_VS.md
├── README.md
└── src
```

Software Package

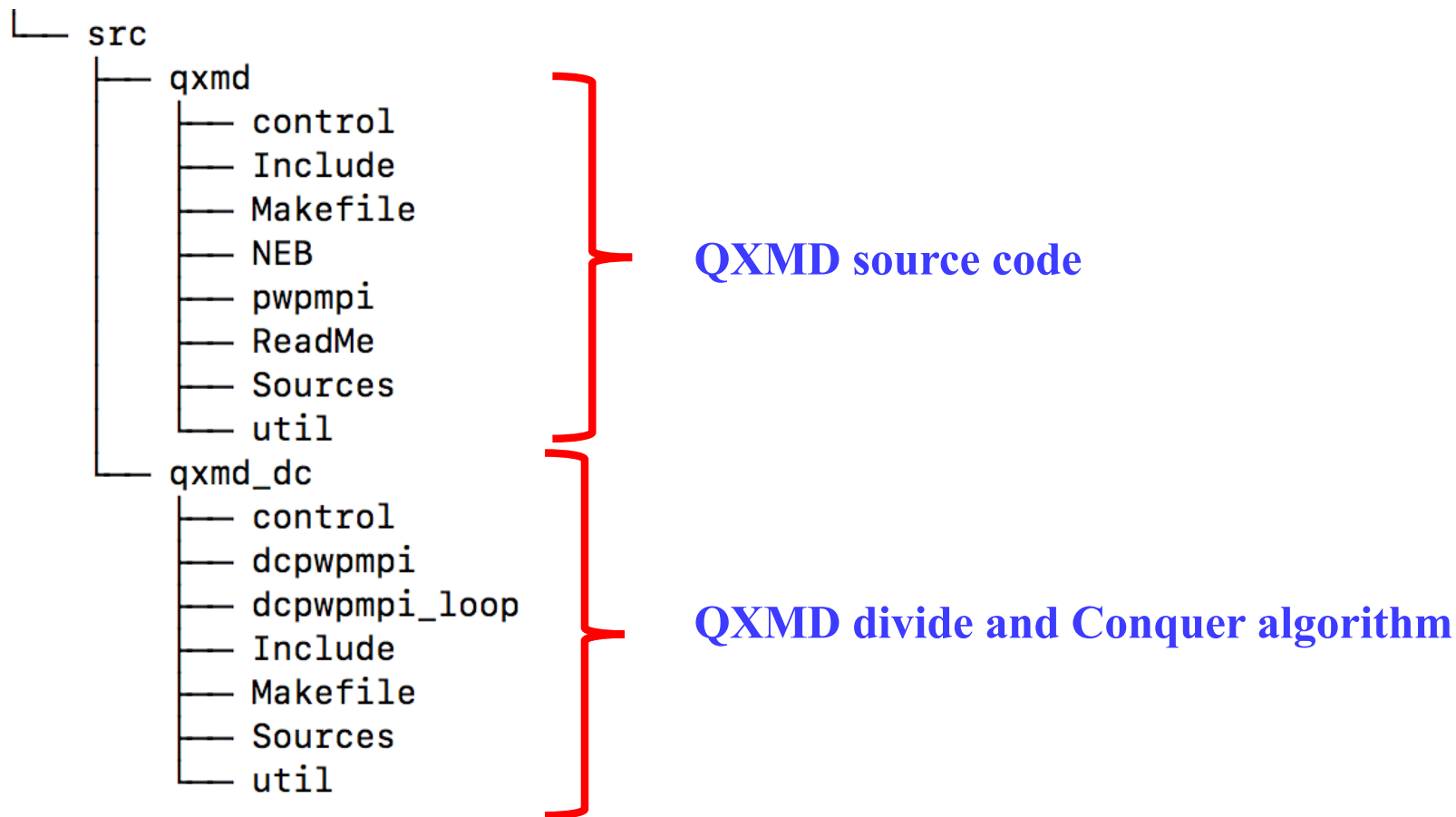
```
•
├── examples
├── QXMD_Manual_VS.md
├── README.md
└── src
```

examples: Example problem QXMD source code

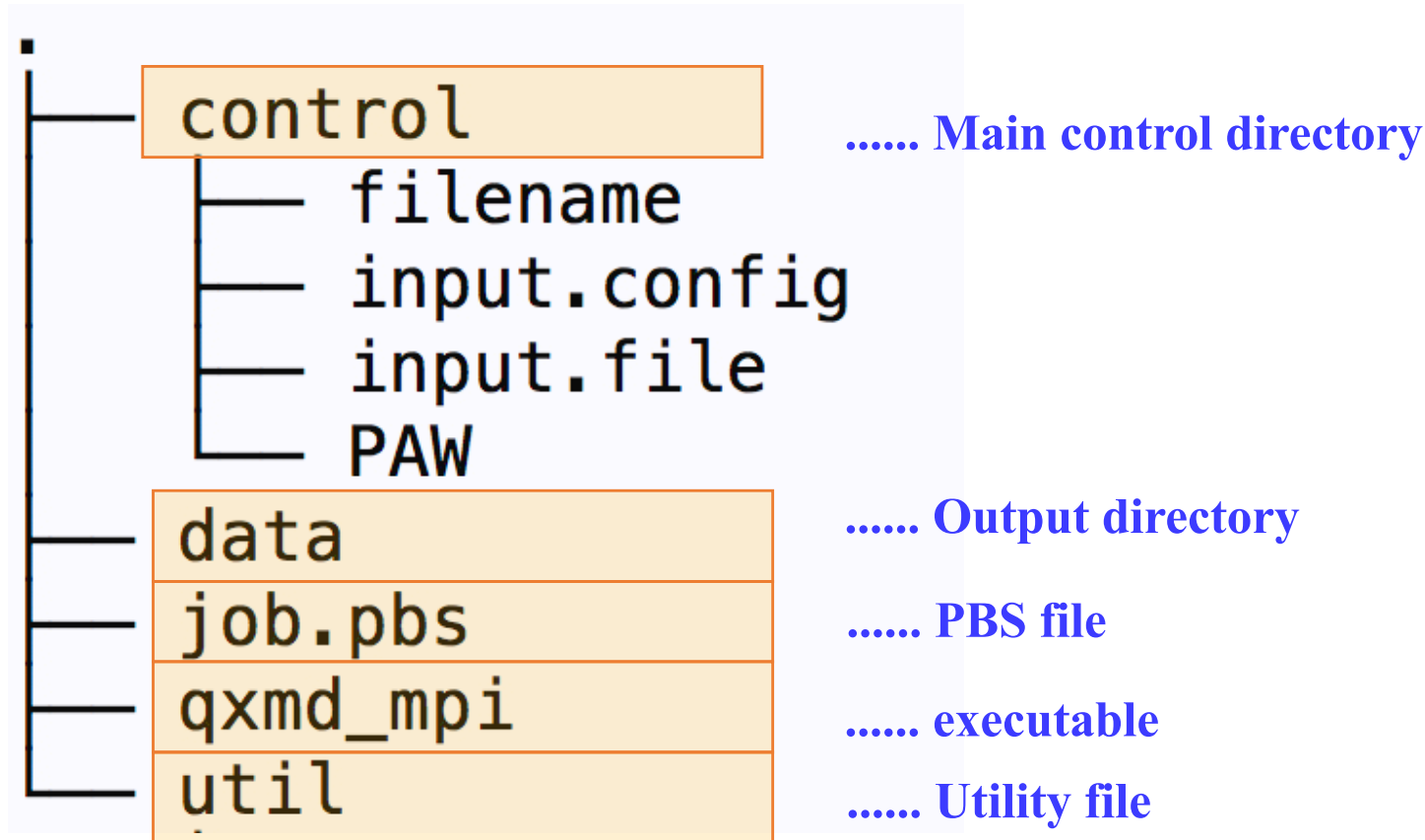
QXMD_Manual_VS.md: Manual and readme

src: Source code

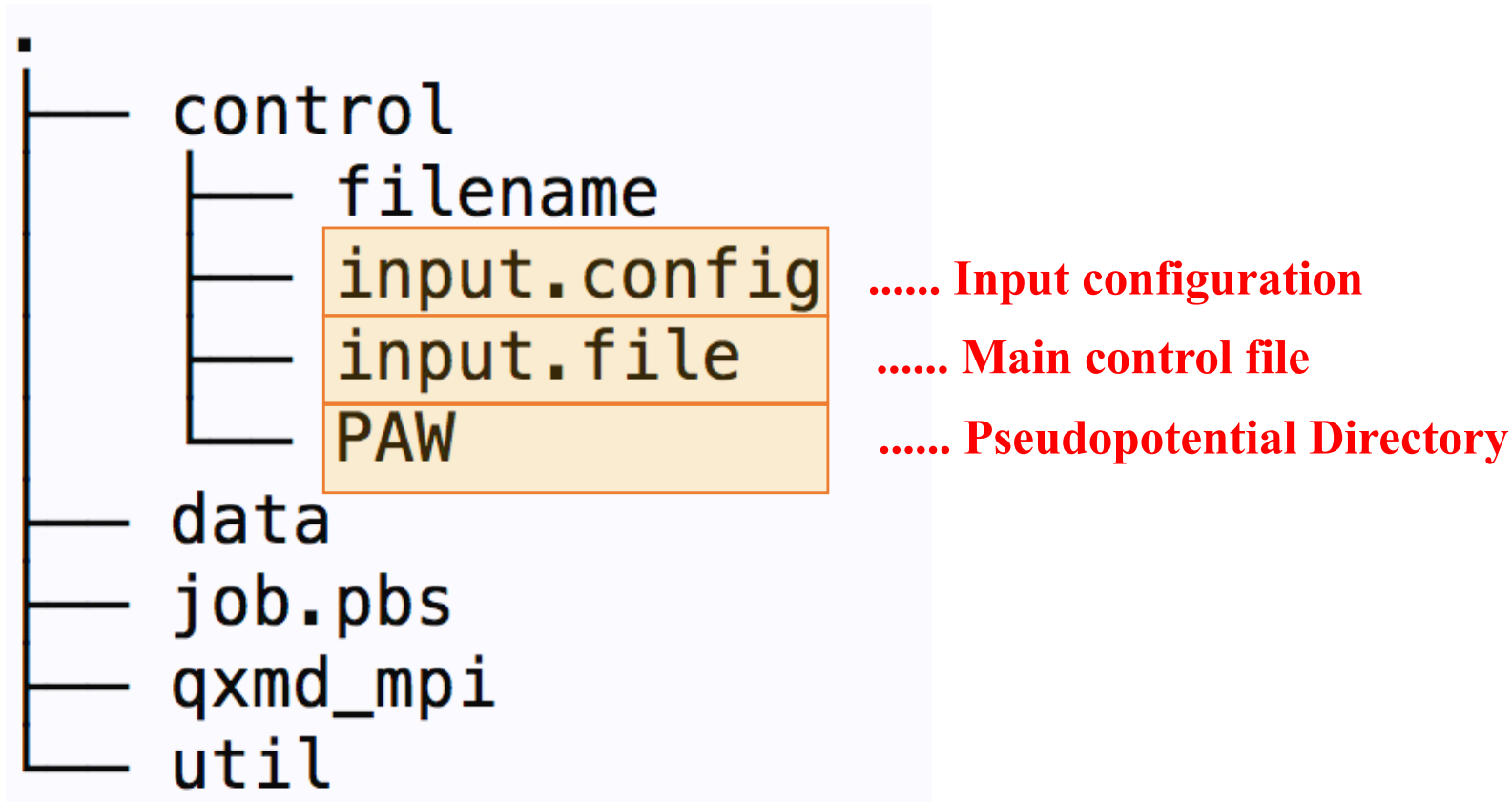
Software Package



Directory Structure



Directory Structure



\$ less input.file

Control Directory

control/PAW/

PAW directory must contain potential file for each atom used in

Example: For H₂O, we have H.PBE and O.PBE

control/input.config

**This file contains ionic positions either in fractional coordinate
or real coordinate**

Example:

Fractional

75

1	0.853	0.625	0.321
1	0.836	0.670	0.415
.....			

Unitless

Real

75

1	0.000	0.000	0.0
2	1.757	-0.586	0.0
.....			

Units are Å or bohr

Input.config

control/input.config

Fractional

75

1	0.853	0.625	0.321
1	0.836	0.670	0.415

.....

Unitless

Real

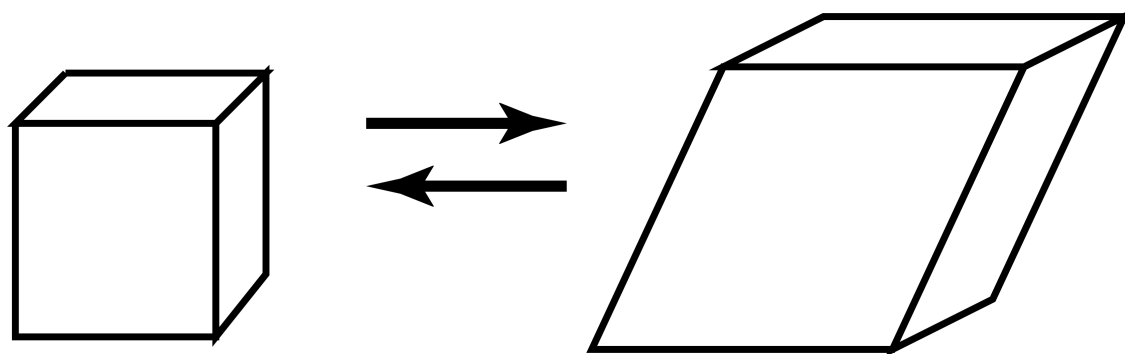
75

1	0.000	0.000	0.0
2	1.757	-0.586	0.0

.....

Units are Å or bohr

Both fractional or real can be given as input



Input.config (water/MoSe₂)

control/input.config

Real

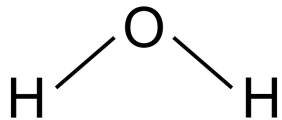
75

1	0.000	0.000	0.0
2	1.757	-0.586	0.0

.....

Units are Å or bohr

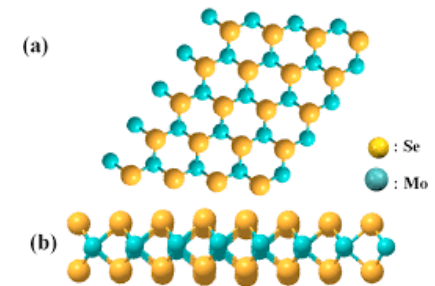
Water:



O- 1
H- 2

MoSe₂

Mo- 1
Se - 2



Input File

control/input.file

Main control file

A Template is provided with your program

Control file is divided into several sections. Each section start with its ***\$SECTION_NAME** and ends with ***end**.

Example

```
*parallel           :  
(QM-nodes)         :  
  1 1 1             : (npx, npy, npz)  
(k-points)         :  
  1                 : (npk)  
(MD-nodes)         :  
  1 1 1             : (md_npx, md_npy, md_npz)  
*end               :
```

Input File: Enable/Disable calculation

Enabling section

Each section name **must start with 1 column** of the file to enable

Disabling section

To disable set **false** at the sub-section (How).

Other option is to entirely delete the section. If it's required program will take a default value.

Mandatory Input: Parallel

```
*parallel           :  
(QM-nodes)          :  
  1 1 1             : (npx, npy, npz)  
(k-points)          :  
  1                 : (npk)  
(MD-nodes)          :  
  1 1 1             : (md_npx, md_npy, md_npz)  
*end               :
```

QM-Nodes: Parallelization over band

K-points: Parallelization over k-points

MD-nodes: Used for divide-conquer-recombine algorithm for order
N DFT code

Mandatory Input: restart/PAW

```
*start           :  
(how of it)      :  
.false.          : (lstart) .true. = restart  
*end            :
```

Set **.true.** , if you would like to restart your job from previous file.
QM_\$file must be present to restart a job.

Mandatory Input: restart/PAW

```
*start           :  
(how of it)      :  
.false.          : (lstart) .true. = restart  
*end            :
```

Set **.true.** , if you would like to restart your job from previous file.
QM_\$file must be present to restart a job.

```
*PAW  
:(how of it)      :  
.true.             : (lpaw) .true. = PAW method  
                   : .false. = pseudopotential method  
*end
```

.true. Projected Augmented Wave method
.false. pseudopotential method

Mandatory Input: Exchange Correlation

```
*approximation for Exc      :  
(approximation)           :  
      2                      : 1:LDA, 2:GGA(PBE)  
(DFT-D)                   :  
.true.                     : (ldftd)  
*end
```

Mandatory Input: Exchange Correlation

```
*approximation for Exc      :  
(approximation)            :  
      2                      : 1:LDA, 2:GGA(PBE)  
(DFT-D)                    :  
      .true.                  : (ldftd)  
*end
```

Approximation

LDA	1
GGA	2
GGA(RPBE)	3
GGA(revPBE)	4
vdW-DF	5
vdW-DF2	6

Mandatory Input: Exchange Correlation

```
*approximation for Exc :  
(approximation) :  
2 : 1:LDA, 2:GGA(PBE)  
(DFT-D) :  
.true. :(ldftd)  
*end
```

Approximation

LDA	1
GGA	2
GGA(RPBE)	3
GGA(revPBE)	4
vdW-DF	5
vdW-DF2	6

Empirical Correction

DFT-D	vdW interaction
DFT-U	Mean field Hubbard model

Mandatory Input: SCF

```
*SCF iteration           :  
(global iteration)      :  
    100                  :  
(tolerance)             :  
    3.0d-08              : (tolerance for total energy)  
    5.0d-08              : (tolerance for average residual)  
*end
```

Tolerance are relative change between two successive run. **Units are in a.u.**

Mandatory Input: Molecular dynamics

***molecular dynamics** :
(how of it) :
1 : (ifmd)

Method

Debug	0
Optimization	1
NVE	2
NVT	3
NPT	4
MSST	10

Mandatory Input: Molecular dynamics

***molecular dynamics** :
(how of it) :
1 : (ifmd)

Method

Debug	0
Optimization	1
NVE	2
NVT	3
NPT	4
MSST	10

(time step) :
0.1d0 1000 : (dtmd, nstop)
time step, total step

Time step is in a. u.

Mandatory Input: Molecular dynamics

(temperature)	: only for real dynamics (NVE-, NVT-, NPT-MD)
300.d0	: (treq) temperature in [K]
(check temperature)	:
.false.	: (liscalc) .true. = Do it !
25	: (iscnum) number of temperature check
20	: (iscstp) skip step

If check temperature is true: First 500 step will have velocity scaling. Since we have set it to false, no velocity scaling will be done

iscnum=Total number of scaling performed

iscstp= scale every iscstp step

Mandatory Input: Molecular dynamics

(temperature)	: only for real dynamics (NVE-, NVT-, NPT-MD)
300.d0	: (treq) temperature in [K]
(check temperature)	:
.false.	: (liscale) .true. = Do it !
25	: (iscnum) number of temperature check
20	: (iscstp) skip step

(optimization)	: only for structural optimization (ifmd == 1)
2	: (ioptmze)

Method

Do not optimize coordinate	-1
Conjugate gradient	0
Projected Velocity Verlet	1
Qausi Newton Method	2

Mandatory Input: Molecular dynamics

(stabilizer for quasi-Newton) :

0.1d0

: (gammamin)

:

(clear Hessian)

:

0

: (ibfgsclear) clear Hessian every ibfgsclear step

Mandatory Input: Molecular dynamics

(stabilizer for quasi-Newton) :

0.1d0 : (gammamin)

:

(clear Hessian)

:

0 : (ibfgsclear) clear Hessian every ibfgsclear step

(tolerance)

: tolerance (ifmd == 1)

1.d-07

: (tol_energy) energy/atom in [a.u.]

5.d-04

: (tol_force) max. force in [a.u.]

***end**

:

Tolerance is in the unit of Hartree and Hartree/bohr

Mandatory Input: Supercell/Cutoff Energy

*supercell	:
(unit of length)	:
(ang)	: (bohr) or (ang)
	:
(lengths & angles)	:
7.00d0, 7.00d0, 5.0d0	: lengths of cell vectors
90.000, 90.000, 90.000	: angles between cell vec. in [deg.]
*end	

Mandatory Input: Supercell/Cutoff Energy

*supercell	:
(unit of length)	:
(ang)	: (bohr) or (ang)
	:
(lengths & angles)	:
7.00d0, 7.00d0, 5.0d0	: lengths of cell vectors
90.000, 90.000, 90.000	: angles between cell vec. in [deg.]
*end	

*planewaves	:
(unit of cutoff energy)	:
(ry)	: (ry) or (hr) or (ev)
(for wavefunctions)	:
30.0	: (ecut)
(for electron density)	:
250.0	: (ecutdens)
(for soft part of density)	:
70.0	: (ecutsoft)
*end	:

Mandatory Input: Supercell/Cutoff Energy

```

*electronic bands
(occupied bands)
  8
(empty bands)
  2
(broadening)
  3  500.d0
*end
  
```

:
 :
 : (noband) No. of occupied bands
 :
 : (neband) No. of empty bands
 : total No.= noband + neband
 :
 :
 : (lfermi) = 1:nonmetallic, 2:Fermi, 3:Gaussian,
 :

$$\text{Min occupied bands} = \frac{\text{No. of electron}}{2} \times 1.1$$

$$\text{Water} = \frac{8}{2} \times 1.1 = 4.4 \cong 5$$

Empty band= 1-20

Unit of smearing is Kelvin

Mandatory Input: atom

*atoms	:
(species)	:
2	: (ntype) No. of atomic species
=====:	
(atomic number)	:
8.0	: (zatom)
(pseudopotential)	:
uspp	: kbpp .or. uspp .or. vand
(nonlocal potential)	:
.true. 1.5d0 1.25d0 0.8d0	: (lking) .true. = on, (rking, gkgmax, gkgexct)
(local potential)	:
.false. 1.5d0 1.15d0 0.8d0	: (llking) .true. = on, (rlking, glkgmax, glkgexct)
(partial core correction)	:
.true. 1.4d0	: (lpcc) .true. = on, (r_cut) in [a.u.]
.true. 1.1d0 1.15d0 0.8d0	: (lpking) .true. = on, (rpking, gpkgmax, gpkgexct)
	: smoothing parameters
	:

Mandatory Input: atom

(unit of length)	: only for positions
(ang)	: (bohr) or (ang)
	:
(position file)	: Ignored, if (nhk) > 0.
'control/input.config'	:
2	: 1:scaled, 2:real coordinates
1	: (keyword)
	:
(fix positions)	:
.false.	: (lfixion) .true. = fix atomic position
	:
(end)	:
*end	

For fix position **.true.**, create a **new atom ID** and set fix position true

Some optional Input: dump

```

*dump charge density      :
(how of it)                :
.true.                     : (ldpchg) .true. = Do it !
(skip step)                 : only for molecular dynamics
5                           : (nskip_dpchg)
(output area)              : output area for charge density
1.0  0.0                   : x_min & x_max
1.0  0.0                   : y_min & y_max
1.0  0.0                   : z_min & z_max
*end                      :
*dump wavefunctions      :
(how of it)                :
.true.                     : (ldpwav) .true. = Do it !
(bands)                    :
79, 85                     : (ibstt1,ibstt2) band index ( 0, 0 -> all bands)
(skip step)                 : only for molecular dynamics
5                           : (nskip_dpwav)
*end                      :
    
```

If ($x_{\min} > x_{\max}$) dump charge density for whole space

Some optional Input: On the fly results

***stress calculation**

(how of it)

.true.

(skip step)

5

***end**

: only for bulk calculations

:

: (lstress) .true. = Do it !

: only for molecular dynamics

: (nskip_stress)

:

:

:

:

: (lintchg) .true. = Do it !

: only for molecular dynamics

: (nskip_intchg)

:

***atomic charge**

(how of it)

.true.

(skip step)

5

***end**

Example:01

```
- examples
├── 01_Water
│   ├── adiabatic_qmd
│   └── optimization
├── 02_MoSe2
│   ├── naqmd
│   └── optimization
├── 03_DPT_linear_response
│   ├── control
│   ├── job.pbs
│   └── qxmd_mpi
├── 04_Si_MSST
│   ├── control
│   ├── job.pbs
│   └── qxmd_mpi
└── 05_Al2O3_pwpdc
    └── input.file
```

Let's head to example 01. We will optimize single water molecule in the box and perform MD

```
$ cd example/01_water/optimization
```


Hands-on I

Copy executable from program directory

```
$ cp QXMD_Course/src/PWP/qxmd_mpi .
```

Submit job using command

```
$qsub job.pbs
```

Hands-on: Visualization

Create PDB file

Copy program to create PDB file to your direcotry

```
$ cp QXMD_Course/src/PWP/util/toPDBcell.f .
```

Compile program

```
$ ifort toPDBcell.f -o toPDBcell
```

Run Program

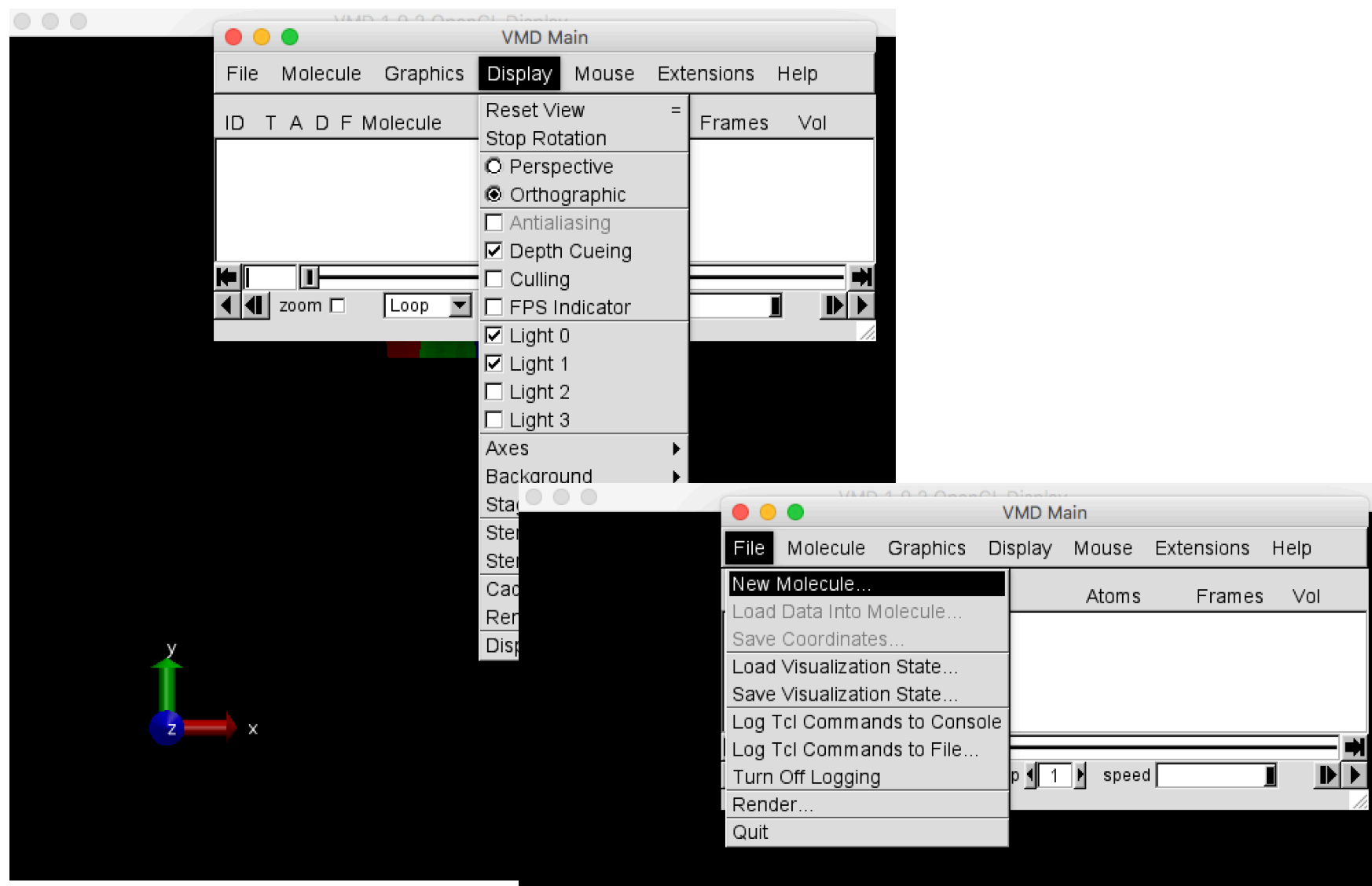
```
$/toPDBcell
```

```
open :  
data/qm_ion.d
```

```
open :  
data/qm_box.d
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Visualization of Output File in VMD



Visualization of Output File in VMD

The screenshot displays the VMD 1.9.4a9 OpenGL Display window. The main visualization area shows a large, flat, sheet-like molecular structure composed of many small, interconnected atoms and bonds, rendered in a stick representation. The structure is primarily green and blue, with some red and orange highlights. A 3D coordinate system is visible in the bottom left corner of the main window, with axes labeled x (red), y (green), and z (blue).

On the left side, a console window titled "sctiwari — startup.command — VMD • startup.comm" displays the following startup logs:

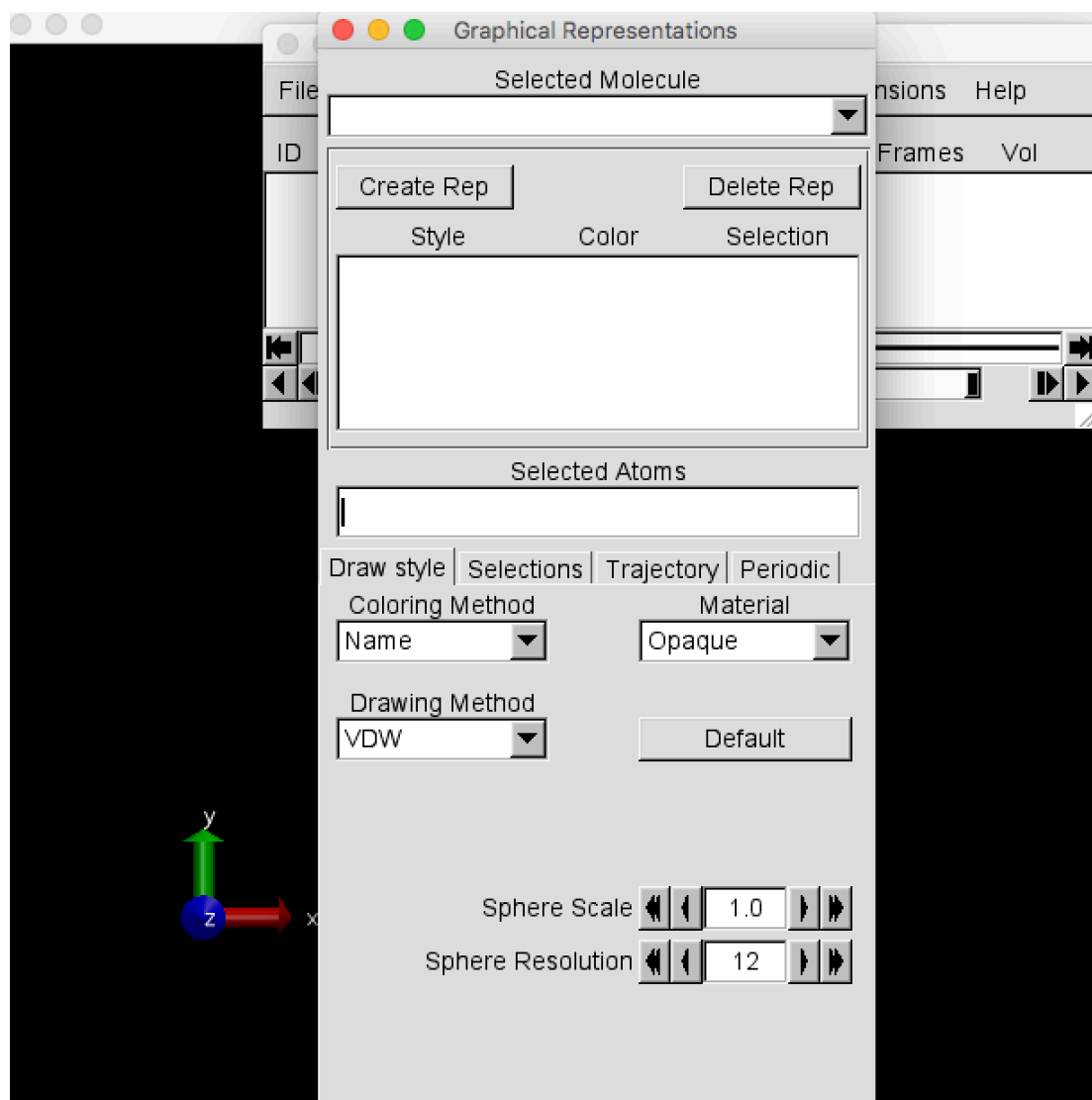
```
Info) -----
Info) Multithreading available, 4 CPUs detected.
Info) OpenGL renderer: Intel(R) Iris(TM) Plus Graphics 6
Info) Features: STENCIL MDE MTX NPOT PP PS GLSL(OVF)
Info) Full GLSL rendering mode is available.
Info) Textures: 2-D (16384x16384), 3-D (2048x2048x2048)
Info) Dynamically loaded 2 plugins in directory:
Info) /Applications/VMD 1.9.4.app/Contents/vmd/plugins/M
vmd > Info) Using plugin xyz for structure file /Users/s
inal_2H.xyz
Info) Using plugin xyz for coordinates from file /Users/
final_2H.xyz
Info) Determining bond structure from distance search ..
Info) Analyzing structure ...
Info) Atoms: 1255
Info) Bonds: 1340
Info) Angles: 0 Dihedrals: 0 Improper: 0 Cross-te
Info) Bondtypes: 0 Angletypes: 0 Dihedraltypes: 0
Info) Residues: 5
Info) Waters: 0
Info) Segments: 1
Info) Fragments: 5 Protein: 90 Nucleic: 0
Info) Finished with coordinate file /Users/sctiwari/Desk
```

Below the console window, the "VMD Main" window shows a menu bar (File, Molecule, Graphics, Display, Mouse, Extensions, Help) and a table of loaded molecules:

ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	kevlar_final_2H.xyz	1255	1	0

At the bottom of the VMD Main window, there is a playback control bar with buttons for "zoom", "Loop", "step", and "speed".

Visualization of Output File in VMD



Visualization of Output File in VMD

The screenshot displays the VMD 1.9.4a9 OpenGL Display window, showing a molecular structure (kevlar_final_2H.xyz) rendered in a 3D view. The structure is a long, thin, repeating unit of a polymer chain, colored in green and blue. Below the main window, the VMD Main window is visible, showing a table of loaded molecules and a console window with output information.

VMD Main

ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	kevlar_final_2H.xyz	1255	1	0

Graphical Representations

Selected Molecule: 0: kevlar_final_2H.xyz

Create Rep Delete Rep

Style	Color	Selection
VDW	Name	all
DynamicBonds	Name	all

Selected Atoms: all

Draw style: Selections | Trajectory | Periodic | Coloring Method: Material | Name: Name | Opaque: Opaque | Drawing Method: DynamicBonds | Default: Default

Distance Cutoff: 1.6 | Bond Radius: 0.3 | Bond Resolution: 12

Apply Changes Automatically Apply

Console Output:

```
Info) Full GLSL rendering mode is available.
Info) Textures: 2-D (16384x16384), 3-D (2048x2048x2048), Multitexture (8)
Info) Dynamically loaded 2 plugins in directory:
Info) /Applications/VMD 1.9.4.app/Contents/vmd/plugins/MACOS
XX86/molfile
vmd > Info) Using plugin xyz for structure file /Users/sctiwari/Desktop/kevlar_final_2H.xyz
Info) Using plugin xyz for coordinates from file /Users/sctiwari/Desktop/kevlar_final_2H.xyz
Info) Determining bond structure from distance search ...
Info) Analyzing structure ...
Info) Atoms: 1255
Info) Bonds: 1340
Info) Angles: 0 Dihedrals: 0 Improper: 0 Cross-terms: 0
Info) Bondtypes: 0 Angletypes: 0 Dihedraltypes: 0 Impropertypes: 0
Info) Residues: 5
Info) Waters: 0
Info) Segments: 1
Info) Fragments: 5 Protein: 90 Nucleic: 0
Info) Finished with coordinate file /Users/sctiwari/Desktop/kevlar_final_2H.xyz.
```

Restart Job and Create HOMO/LUMO

```
*start(on/off)      :  
(how of it)         :  
  .false.            : (lstart) .true. = restart  
*end                 :  
                     :
```

1- set start section in input.file to be **.true.**

2- If you desire you can save data directory with some other name. New run will **overwrite** data directory

Creating HOMO and LUMO

Copy following section from **src/qxmd/control/input.file**
to our input.file at **example/01_water/optimization/control/input.file**

```
*dump wavefunctions
(how of it)
.false.
(bands)
0, 0
(skip step)
5
(output area)
1.0 0.0
1.0 0.0
1.0 0.0
*end
```

```
:
:
:
: (ldpwav) .true. = Do it !
:
: (ibstt1,ibstt2) band index ( 0, 0 -> all bands)
: only for molecular dynamics
: (nskip_dpwav)
: output area for charge density
: x_min & x_max Note: In SCALED coordinates.
: y_min & y_max Note: Whole space
: z_min & z_max if xyz_min > xyz_max.
:
:
```

- 1- set dump wavefunction section in input.file to to **.true.**
- 2- set bands to **number of band corresponding to HOMO and LUMO**
- 3- set desired **skip step**. We will set it to 1.

Creating HOMO and LUMO

Copy gcube4.f90 program from util

```
$ cp QXMD_Course/src/PWP/util/gcube4.f90 .
```

Compile program

```
$ ifort gcube4.f90 -o gcube
```

Run Program

Visualization of CUBE File in VMD

The screenshot displays the VMD 1.9.4a9 OpenGL Display window, showing a molecular structure (Kevlar) rendered in a 3D view. The structure is composed of multiple parallel chains of atoms, colored by element (Carbon: grey, Oxygen: red, Nitrogen: blue, Hydrogen: white). The VMD Main window is visible below the 3D view, showing the File menu, a table of loaded molecules, and a timeline for the selected molecule. The Graphical Representations window is also visible, showing the selected molecule and its representation settings.

VMD Main

ID	T	A	D	F	Molecule	Atoms	Frames	Vol
0	T	A	D	F	kevlar_final_2H.xyz	1255	1	0

Graphical Representations

Selected Molecule: 0: kevlar_final_2H.xyz

Create Rep Delete Rep

Style	Color	Selection
VDW	Name	all
DynamicBonds	Name	all

Selected Atoms: all

Draw style | Selections | Trajectory | Periodic | Coloring Method | Material

Name: [Dropdown] Material: Opaque [Dropdown]

Drawing Method: DynamicBonds [Dropdown] Default [Button]

Distance Cutoff: [Slider] 1.6 [Buttons]

Bond Radius: [Slider] 0.3 [Buttons]

Bond Resolution: [Slider] 12 [Buttons]

Apply Changes Automatically [Checkbox] Apply [Button]

Console Output:

```
Info) Full GLSL rendering mode is available.
Info) Textures: 2-D (16384x16384), 3-D (2048x2048x2048), Multitexture (8)
Info) Dynamically loaded 2 plugins in directory:
Info) /Applications/VMD 1.9.4.app/Contents/vmd/plugins/MACOS
XX86/molfile
vmd > Info) Using plugin xyz for structure file /Users/sctiwar
ari/Desktop/kevlar_final_2H.xyz
Info) Using plugin xyz for coordinates from file /Users/scti
wari/Desktop/kevlar_final_2H.xyz
Info) Determining bond structure from distance search ...
Info) Analyzing structure ...
Info) Atoms: 1255
Info) Bonds: 1340
Info) Angles: 0 Dihedrals: 0 Improper: 0 Cross-terms: 0
Info) Bondtypes: 0 Angletypes: 0 Dihedraltypes: 0 Impr
opertypes: 0
Info) Residues: 5
Info) Waters: 0
Info) Segments: 1
Info) Fragments: 5 Protein: 90 Nucleic: 0
Info) Finished with coordinate file /Users/sctiwar/Desktop/
kevlar_final_2H.xyz.
```

Hands-on II: Optimization of MoSe₂

- Change directory to mose2
\$ cd ../mose2
- Find the difference between **mose2 input.file** and **water input.file**

Hands-on II: Optimization of MoSe₂

- Change directory to mose2
\$ cd ../mose2
- Find the difference between **mose2 input.file** and **water input.file**

We have changed the description of atom in atom section

First atom corresponds to Mo

Second atom corresponds to Se

Hands-on II: Optimization of MoSe₂

- Change directory to mose2
\$ cd ../mose2
- Find the difference between **mose2 input.file** and **water input.file**

We have changed the description of atom in atom section

First atom corresponds to Mo

Second atom corresponds to Se

Copy executable from program directory

```
$ cp ../../program/qxmd_mpi .
```

Submit job using command

```
$qsub job.pbs
```

Acknowledgement

[A divide-conquer-recombine algorithmic paradigm for large spatiotemporal quantum molecular dynamics simulations](#)

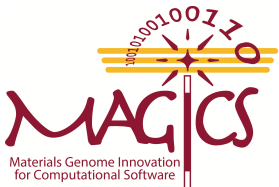
F. Shimojo, R. K. Kalia, M. Kunaseth, A. Nakano, K. Nomura, S. Ohmura, K. Shimamura and P. Vashishta, Journal of Chemical Physics 140, 18A529 (2014).

Materials software (QXMD) used in this research was produced by USC MAGICS Center that is a part of the Computational Materials Sciences Program funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, under Award Number *DE-SC00014607*.



End of Section I

**Thank you for your
attention**



Download (wget)

Download from **MAGICS** website

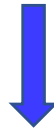
<https://magics.usc.edu/pwp/>

Download using wget

```
$ wget https://github.com/USCCACS/PWP_download/blob/master/PWP-CentOS7.3.1611Core-ibfort16.0.0-OpenMPI1.8.8.tar?raw=true
```

Rename filename

PWP-CentOS7.3.1611Core-ibfort16.0.0-OpenMPI1.8.8.tar?raw=true



PWP-CentOS7.3.1611Core-ibfort16.0.0-OpenMPI1.8.8.tar.