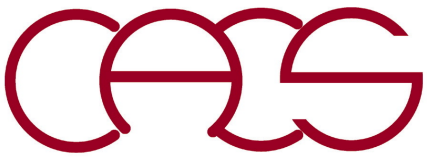
QXMD Hands On: Born-Oppenheimer Molecular Dynamics

Aiichiro Nakano & Taufeq Razakh

Collaboratory for Advanced Computing & Simulations
Department of Computer Science
Department of Physics & Astronomy
Department of Quantitative & Computational Biology
University of Southern California

Email: (anakano, razakh)@usc.edu



Outline

1- Optimization of Geometry

- **Hands-on I: Optimization of a water molecule**

2- Born-Oppenheimer Molecular Dynamics (BOMD)

- **Hands-on II: BOMD simulation of a water molecule**

Project Structure

```
.
├── CMake
├── CMakeLists.txt
...
.
├── build
│   ├── bin
│   │   └── qxmd
├── Example
├── LIB
│   ├── PAW
│   └── NCPP
├── Sources
└── util
```

qxmd: Executable

Example: Learn about using the fundamentals of QMD through practical applications

LIB: Pseudopotential

- **PAW:** Projector Augmented Wave Pseudo-potential
- **NCPP:** Norm Conserving Pseudo-potential

Sources: Source code essential to QXMD

util: Utility source code

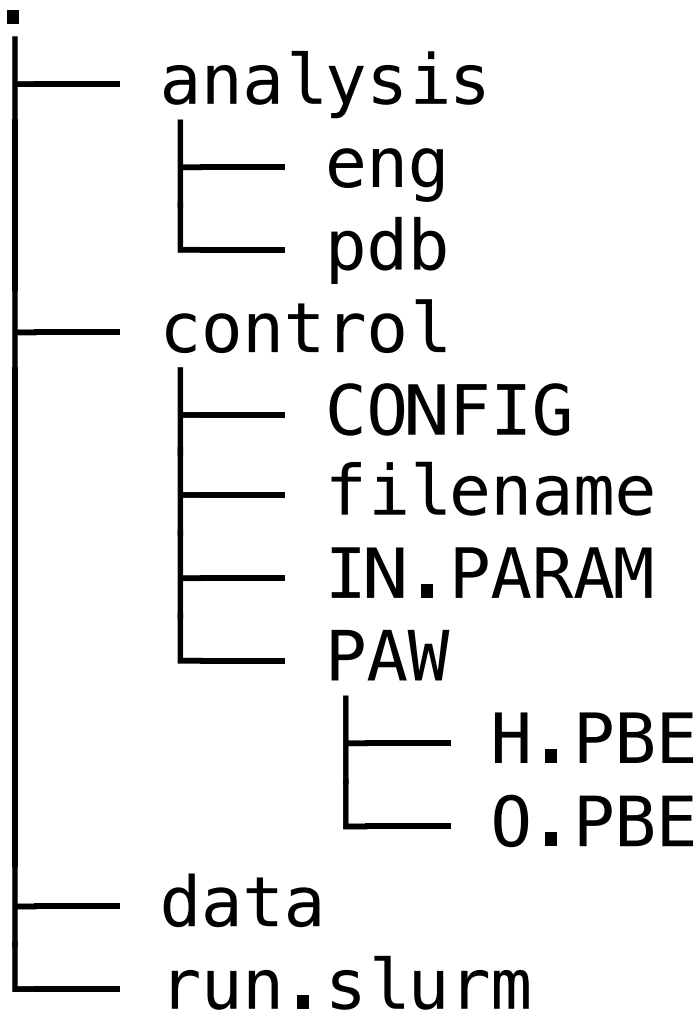
Learning Directory Structure

```
[QXMD_DEV]$ cd Example/
```

```
.
├── 01_Optimization
├── 02_Adaibatic_MD
├── 04_MSST
├── 05_NAQMD
└── 06_LinearResponseTDDFT
```

```
[Example]$ cd 01_Optimization/01_Water
```

Optimization Directory Structure



..... Analysis directory (optional)

..... Main control directory (mandatory)

..... Configuration of structure (mandatory)

..... Main control file (mandatory)

..... Pseudopotential directory (mandatory)
PAW directory should be replaced with USPP
for ultra soft Pseudopotential

..... Output data directory(mandatory) `mkdir data`

..... Script to submit job (mandatory)

Control Directory

control/PAW/

PAW directory must contain potential file for each atom used in

Example: For H_2O, we have H.PBE and O.PBE

control/CONFIG

$ less 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

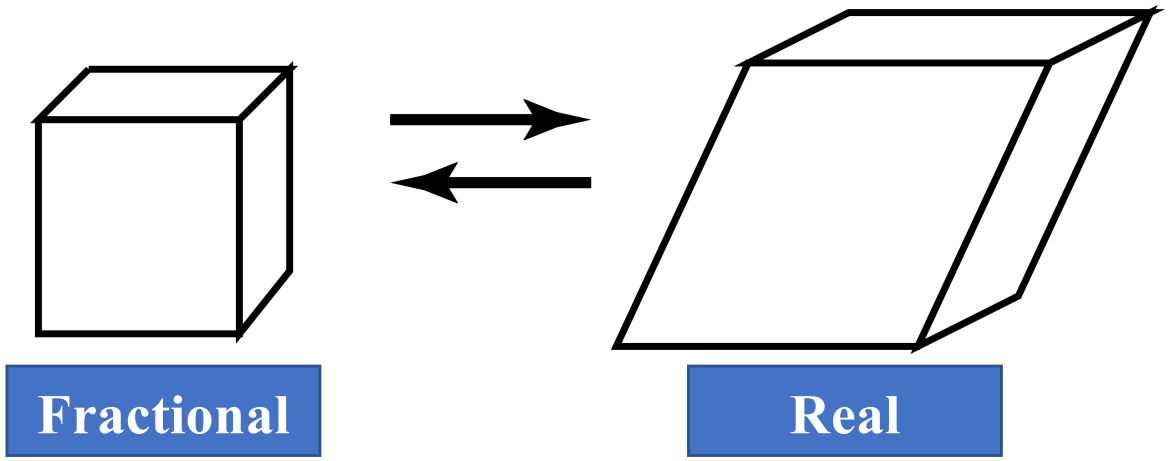
```
75
1     0.000   0.000   0.0
2     1.757   -0.586   0.0
.........
```

Units are Å or bohr

CONFIG

control/CONFIG

Both fractional or real can be given as input



CONFIG (Water/$MoSe_2$)

control/CONFIG

Real

75

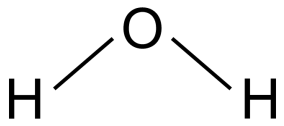
1 0.000 0.000 0.0

2 1.757 -0.586 0.0

……..

Units are Å or bohr

Water:



O- 1

H- 2

IN.PARAM

$ less IN.PARAM

control/IN.PARAM **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
```

SCF: Self consistent field

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.                   : (liscale) .true. = Do it !
  25                       : (iscnum) total 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
Quasi 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                     : (noband)  No. of occupied bands
(empty bands)              :
     2                     : (neband)  No. of empty bands
                           :         total No.= noband + neband
(broadening)               :
     3    500.d0           : (lfermi) = 1:nonmetallic, 2:Fermi, 3:Gaussian,
 *end                      :
```

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

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

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          : only for bulk calculations
(how of it)                  :
 .true.                                  : (lstress) .true. = Do it !
(skip step)                  : only for molecular dynamics
  5                          : (nskip_stress)
*end                         :
                             :
*atomic charge               :
(how of it)                  :
 .true.                      : (lintchg) .true. = Do it !
(skip step)                  : only for molecular dynamics
  5                          : (nskip_intchg)
*end                         :
```

Job Submission 1: Optimization

Submitting jobs to Discovery can be easily done with the `sbatch` command

```
$ pwd
~/QXMD_DEV/Example/01_Optimization/01_Water
$ ls // data directory needs be created before submitting a job
analysis  control  data  run.slurm
$ sbatch run.slurm
```

```
$ cat run.slurm
#!/bin/bash

#SBATCH --account=anakano_429
#SBATCH --partition=main
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem=32G
#SBATCH --time=00:10:00

module load usc
module load fftw

srun --mpi=pmix_v2 -n 1 ../../../build/bin/qxmd
```

Output files

Go to data directory

```
$ cd data
```

Output files: qm_ion.d

Comment

```
#  Atomic scaled coordinates
      0      2      1      2
1.0000000E-01
4.28571 4.28571 7.00000 5.71429 4.28571 6.80000 3.00000 4.28571 6.80000
      1      2      1      2
1.0000000E-01
4.29596 4.28571 7.00366 5.71100 4.28571 6.79902 2.99305 4.28571 6.79732
      2      2      1      2
```

Output files

Go to data directory

```
$ cd data
```

Output files: qm_ion.d

```
#  Atomic scaled coordinates
      0      2      1      2
 1.0000000E-01
 4.28571 4.28571 7.00000 5.71429 4.28571 6.80000 3.00000 4.28571 6.80000
      1      2      1      2
 1.0000000E-01
 4.29596 4.28571 7.00366 5.71100 4.28571 6.79902 2.99305 4.28571 6.79732
      2      2      1      2
```

Step number, No of atom type, Atom type 1, Atom type 2

Output Files

Go to data directory

```
$ cd data
```

Output files: qm_ion.d

```
#  Atomic scaled coordinates
      0      2      1      2
 1.0000000E-01
 4.28571 4.28571 7.00000 5.71429 4.28571 6.80000 3.00000 4.28571 6.80000
      1      2      1      2
 1.0000000E-01
 4.29596 4.28571 7.00366 5.71100 4.28571 6.79902 2.99305 4.28571 6.79732
      2      2      1      2
```

Scaling factor for position of each atoms

Output Files

Go to data directory

```
$ cd data
```

Output files: qm_ion.d

```
#  Atomic scaled coordinates
      0      2      1      2
 1.0000000E-01
 4.28571 4.28571 7.00000 5.71429 4.28571 6.80000 3.00000 4.28571 6.80000
      1      2      1      2
 1.0000000E-01
 4.29596 4.28571 7.00366 5.71100 4.28571 6.79902 2.99305 4.28571 6.79732
      2      2      1      2
```

Coordinate of each atom laid out in x, y, z

Output Files

Output files: qm_box.d

```
#  supercell (FFT cell) vectors (lengths & angles)
#        L_1            L_2            L_3          angle(2-3) angle(3-1) angle(1-2)
       0 1.3228082E+01 1.3228082E+01 9.4486299E+00  90.000000  90.000000  90.000000
```

Comment

Box length in bohr (a.u.)

Output Files

Output files: qm_ion.d

```
#  supercell (FFT cell) vectors (lengths & angles)
#       L_1            L_2            L_3          angle(2-3) angle(3-1) angle(1-2)
      0 1.3228082E+01 1.3228082E+01 9.4486299E+00  90.000000  90.000000  90.000000
```

Comment

Box length in bohr (a.u.)

Output files: md_eng.d

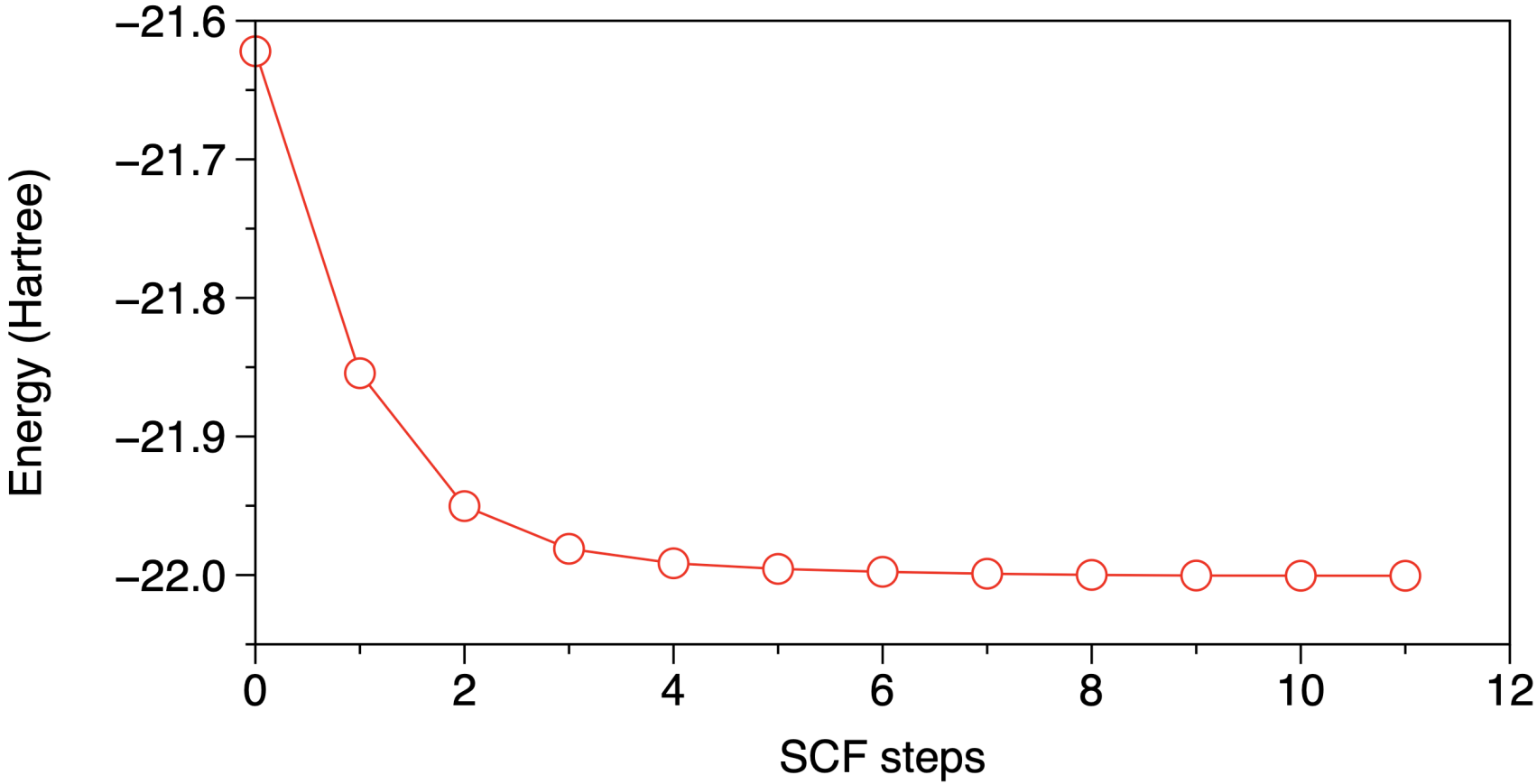
```
# step    P.E. [hartree]
      0 -2.1951549312E+01
      1 -2.1954246118E+01
      2 -2.1959473771E+01
      3 -2.1961990569E+01
      4 -2.1972379455E+01
      5 -2.1978050971E+01
      6 -2.1983590094E+01
      7 -2.2000540405E+01
      8 -2.2001983644E+01
      9 -2.2002236635E+01
     10 -2.2002297278E+01
     11 -2.2002316458E+01
```

Comment

Step and energy (hartree)

Analysis: Energy Convergence

Use a plotting software and plot the energy *vs.* self-consistent field (SCF) iteration steps in md_eng.d



Job Submission 2: BOMD

Now ready to do BOMD simulation

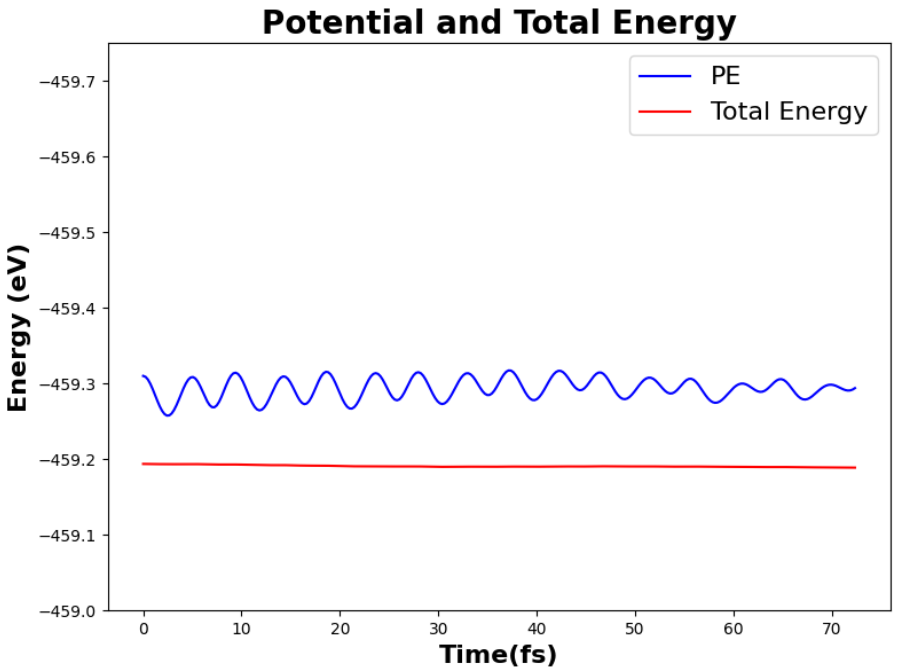
```
$ pwd
~/QXMD_DEV/Example/02_Adaibatic_MD/01_Water_NVE
$ ls // data directory needs be created before submitting a job
control  data  run.slurm
$ sbatch run.slurm
```

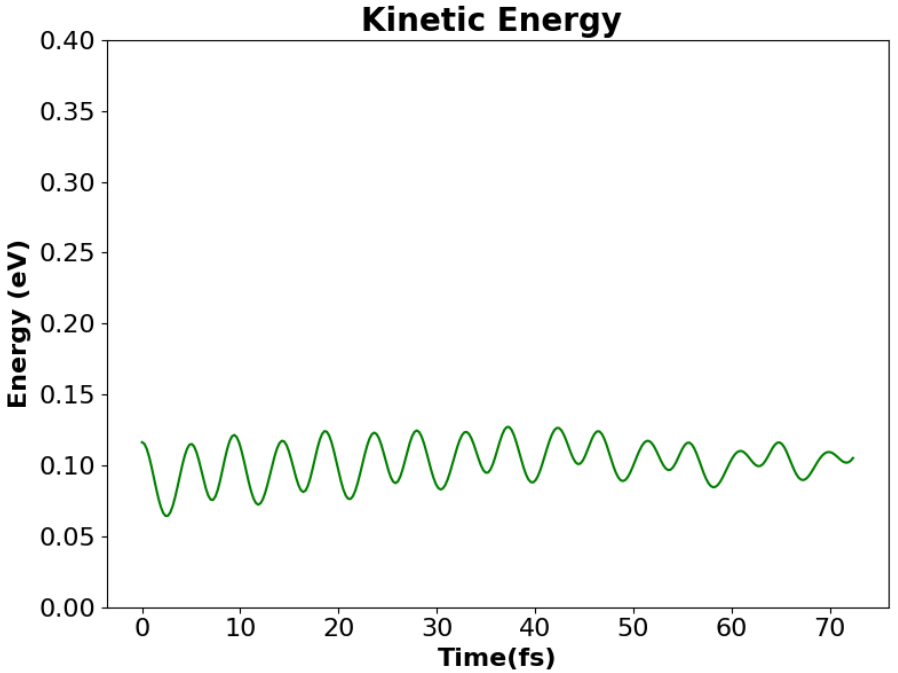
```
$ cat run.slurm
#!/bin/bash


#SBATCH --account=anakano_429
#SBATCH --partition=main
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem=32G
#SBATCH --time=00:10:00


module load usc
module load fftw

cp ../../01_Optimization/01_Water/data/QM_* ./data/
cp ../../01_Optimization/01_Water/data/MD_* ./data/


srun --mpi=pmix_v2 -n 1 ../../../build/bin/qxmd
```

Utilizes optimized water for this simulation

Analysis: Energy vs. Time

Use a plotting software and plot the Kinetic-Potential-Total energy *vs*. Time by looking into md_eng.d





Analysis: Visualization

Go to util directory in your QXMD project directory, locate `toPDBcell.f` & compile

```
$ pwd
~/QXMD_DEV/util
[discovery1 util]$ ls
eig.f  gcube.f90  pick_config2.f90  toPDBcell.f
[discovery1 util]$  gfortran toPDBcell.f -o toPDBcell
```

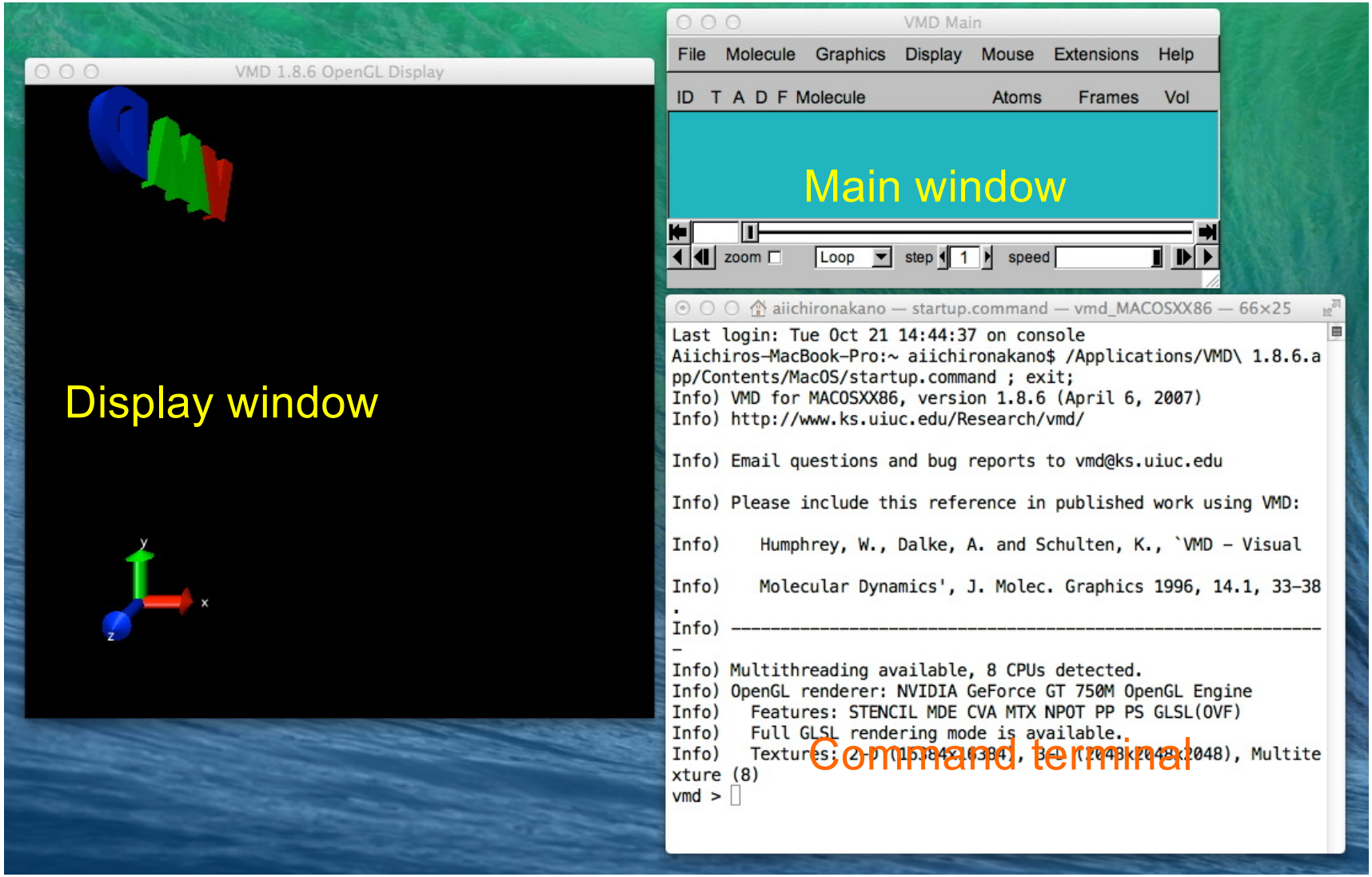
Run program from where the data/ dir is visible

```
$ ls
config.pdb  control  data $ ../../../util/toPDBcell
```

This will generate a `config.pdb` file

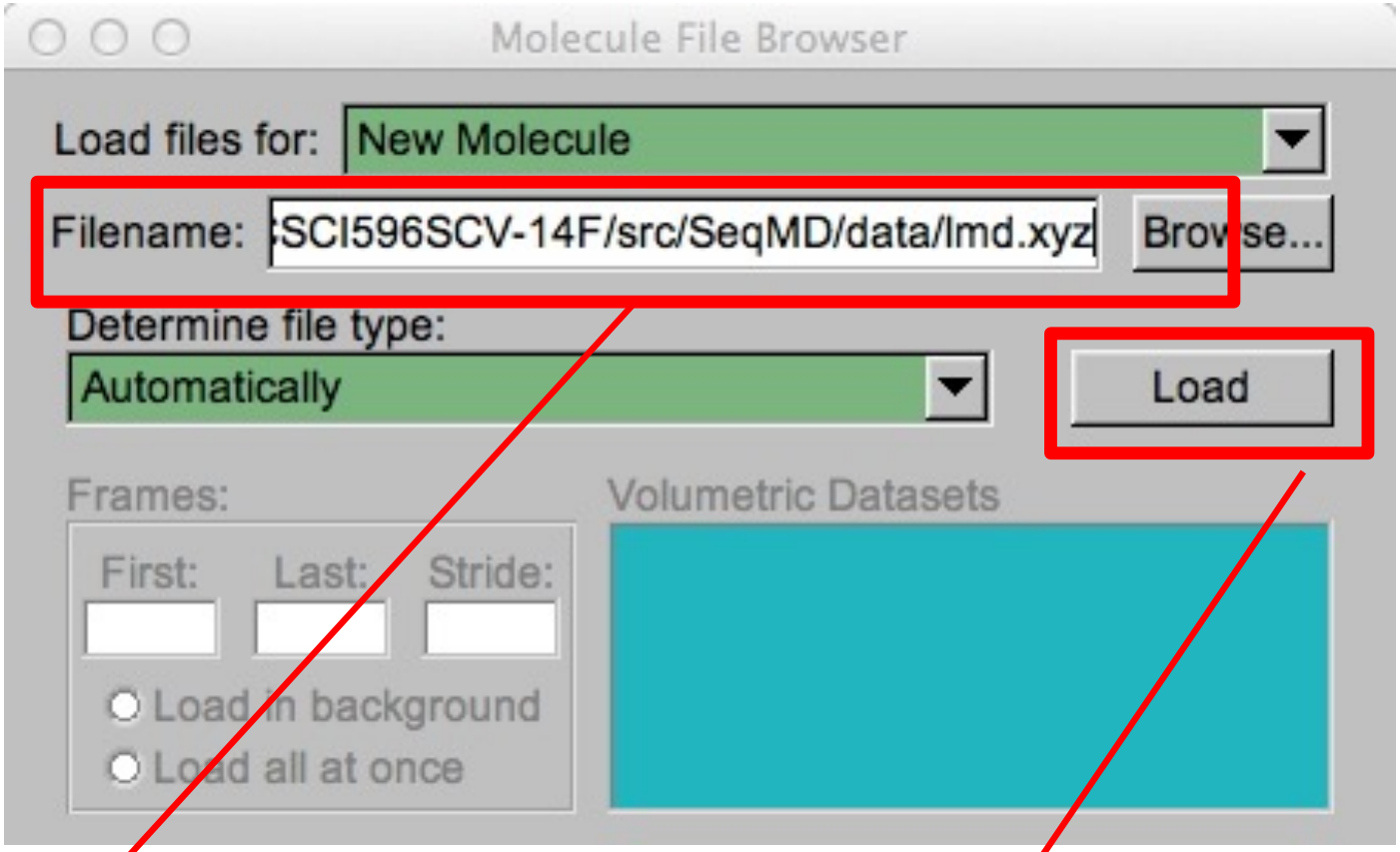
Start VMD

- It will open 3 windows



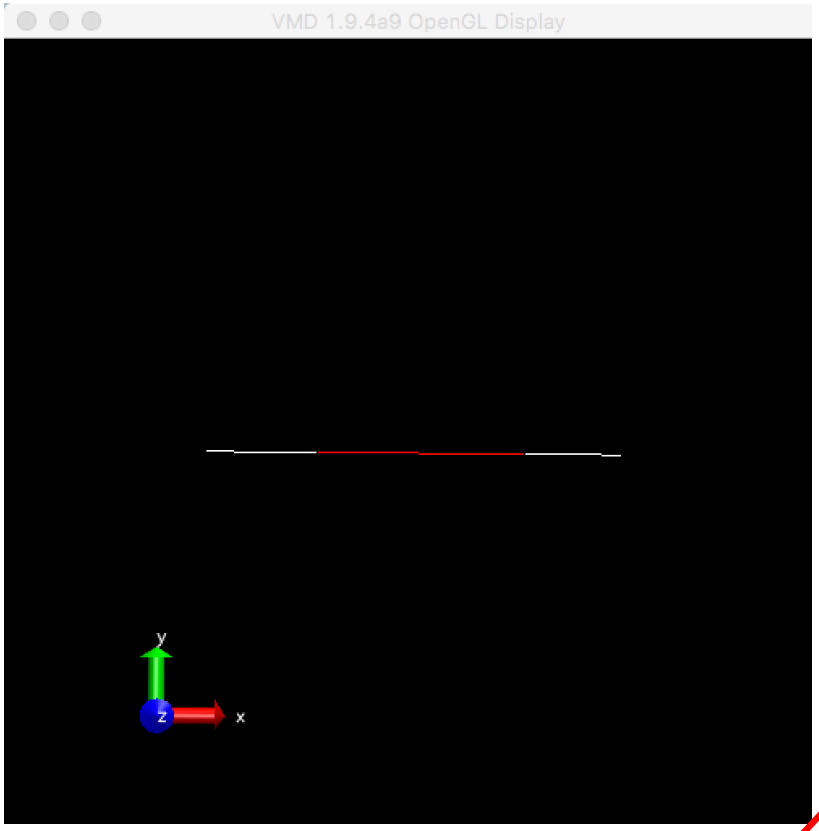
Load the MD-Trajectory PDB File

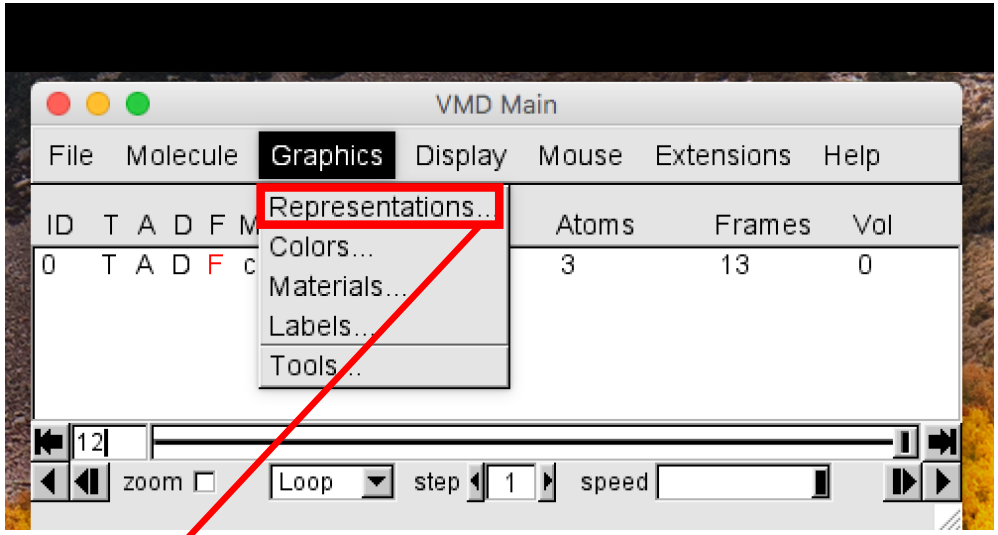
- In the File menu in the VMD main window, select New Molecule; the following new window will open.



- Drag and drop the XYZ file you have created in the Filename field (or press the browse button to locate the file).
- Click the Load button to load the file.

Load the MD-Trajectory PDB File

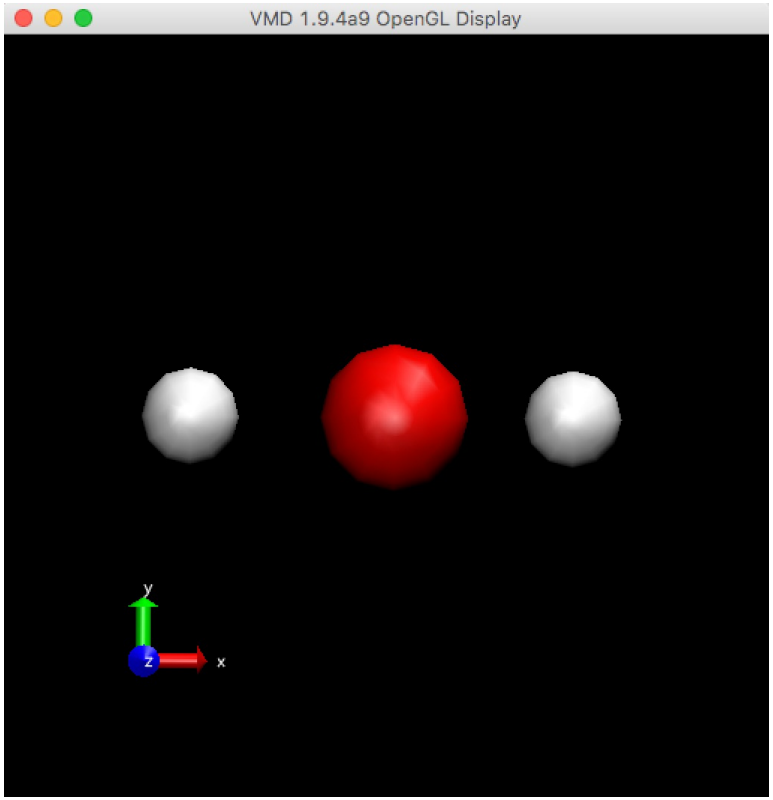




- Click on Representation to make add different representation of each atoms

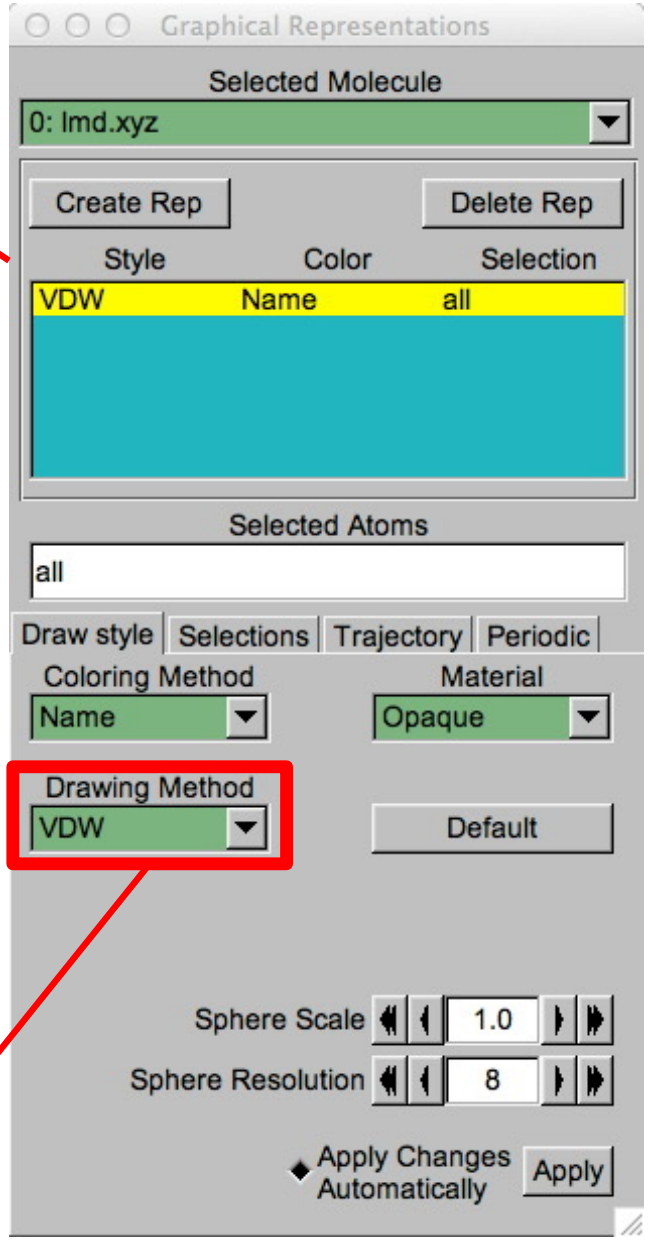
Choose the Graphic Representation

- In the Graphics menu in the VMD main window, select Representations; the following new window will open.



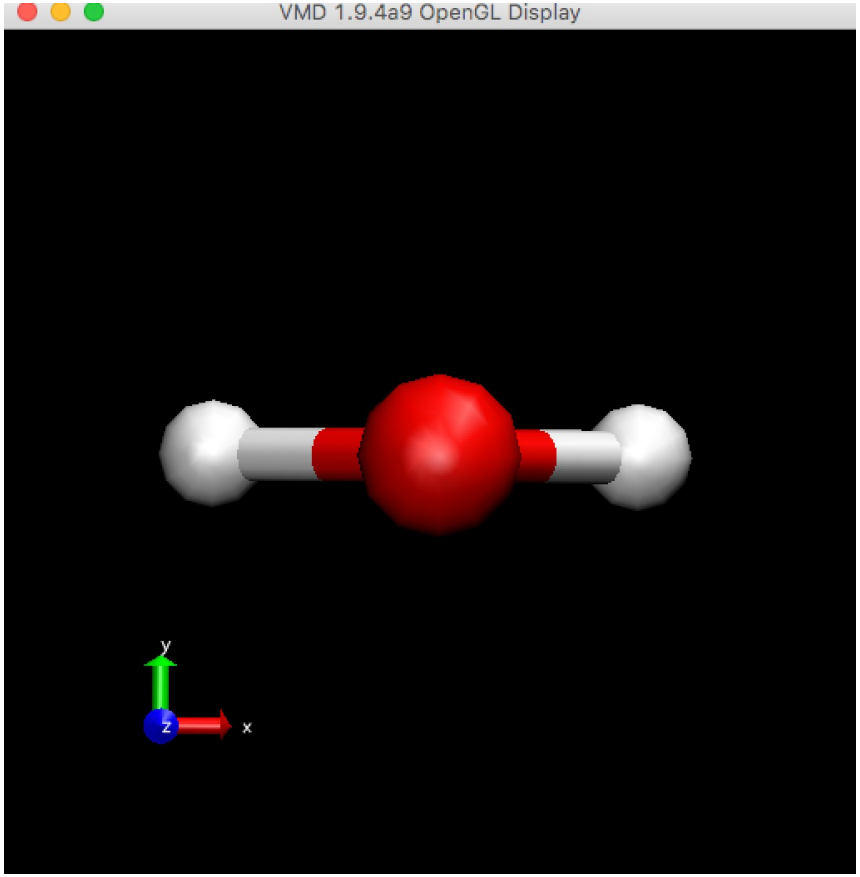
Display now looks like this

- In the Drawing Method menu, choose the VDW (van der Waals radius) representation.



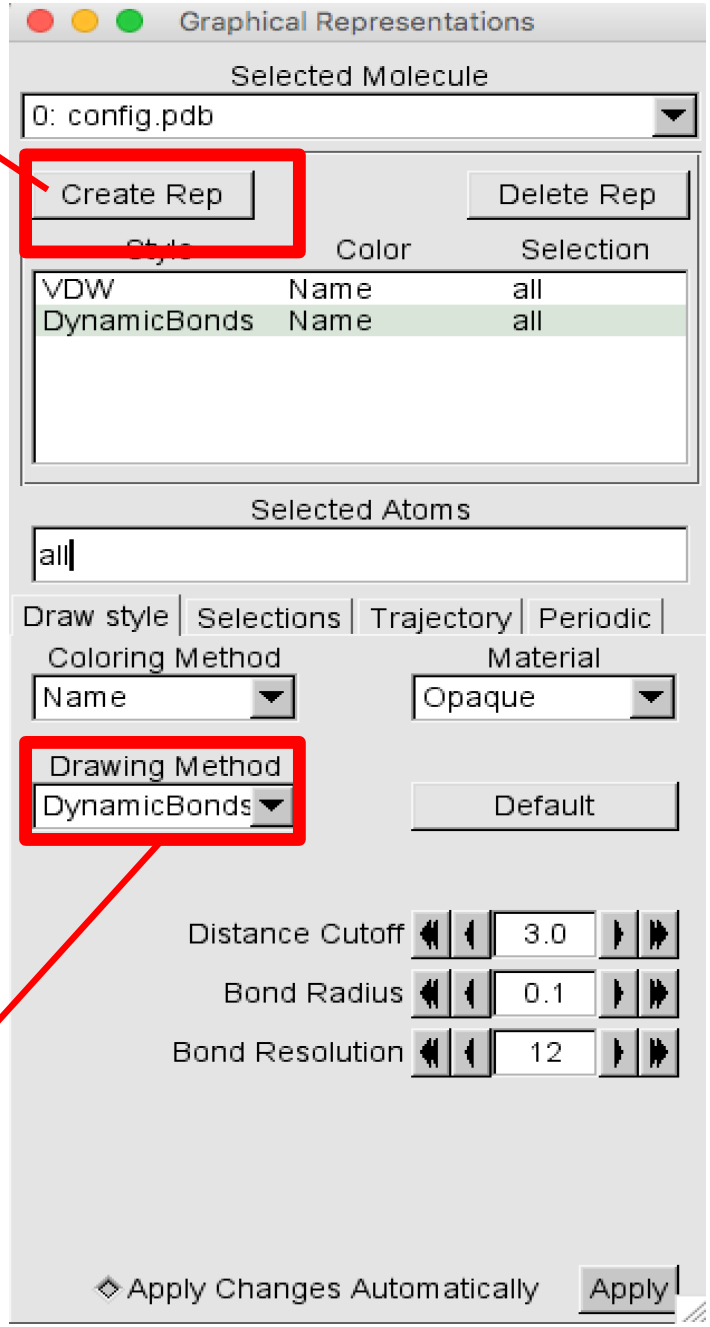
Choose the Graphic Representation

- In the Graphical Representations window; click on Create Rep

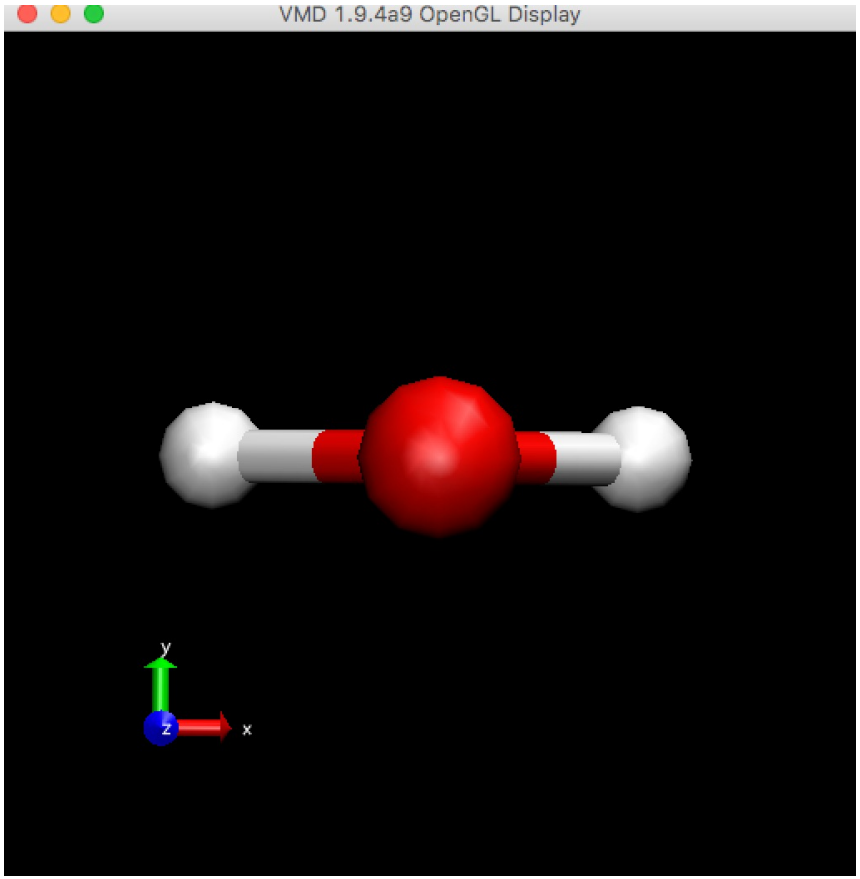


Display now looks like this

- In the Drawing Method menu, choose the DynamicBonds representation.

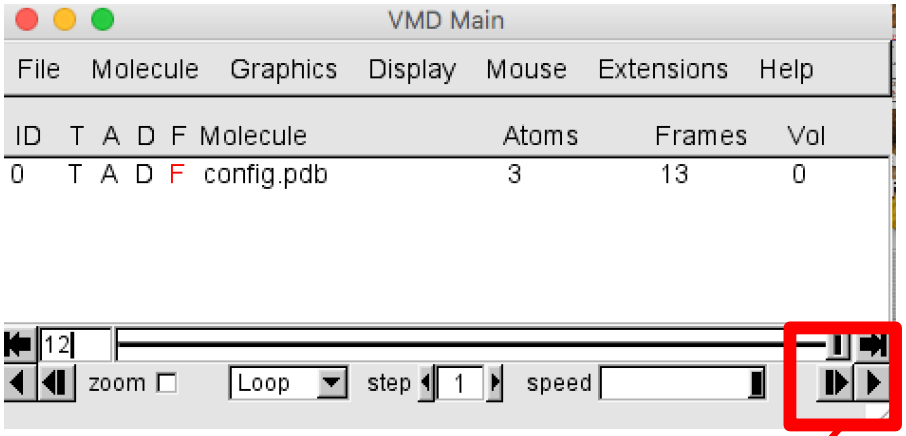


Play Movie



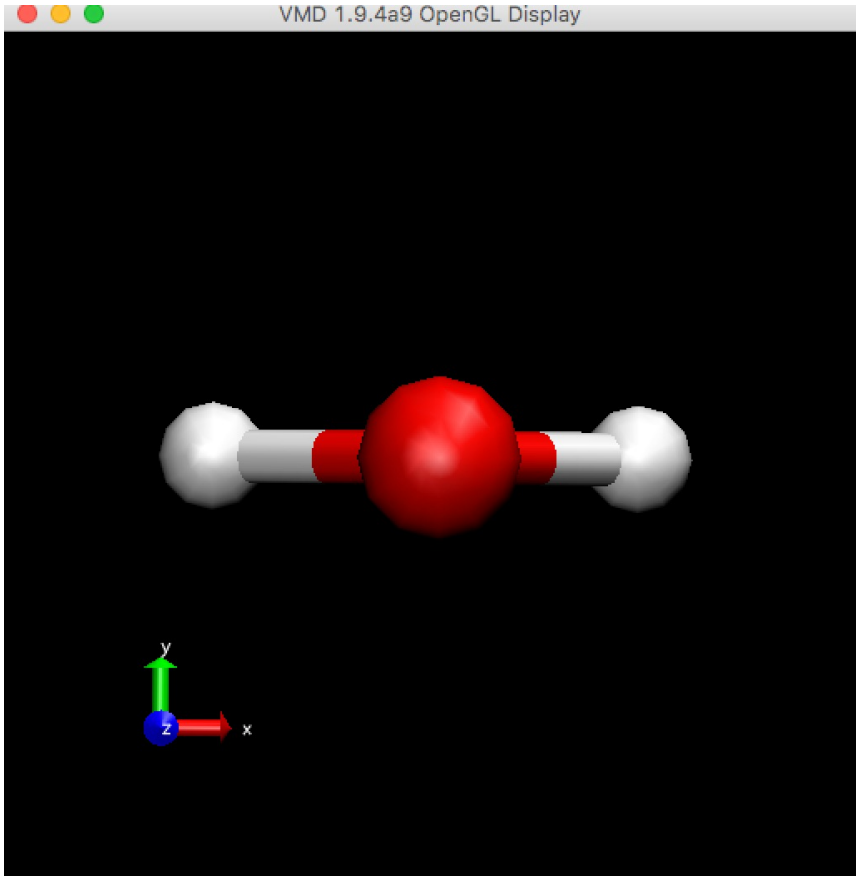
Display now looks like this

- In the Graphical Representations window; click on Create Rep



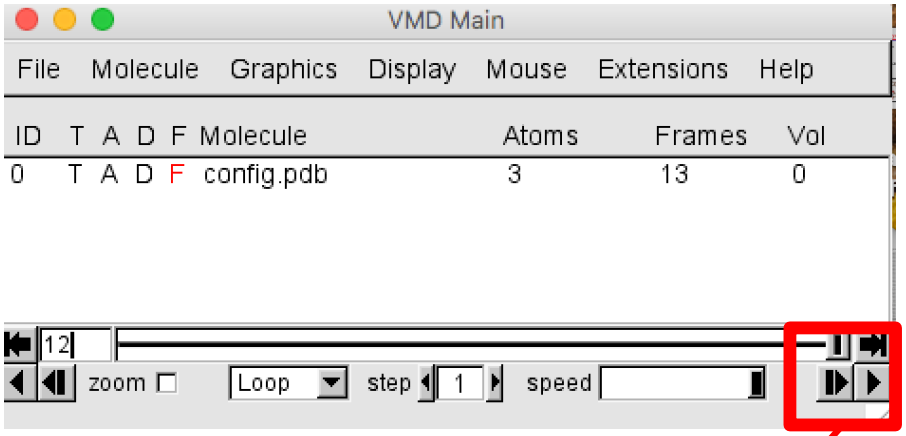
- Play movie by clicking on 
- Watch frame by frame 

Play Movie



Display now looks like this

- In the Graphical Representations window; click on Create Rep



- Play movie by clicking on
- Watch frame by frame movie