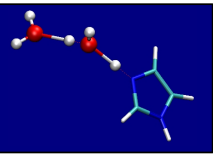


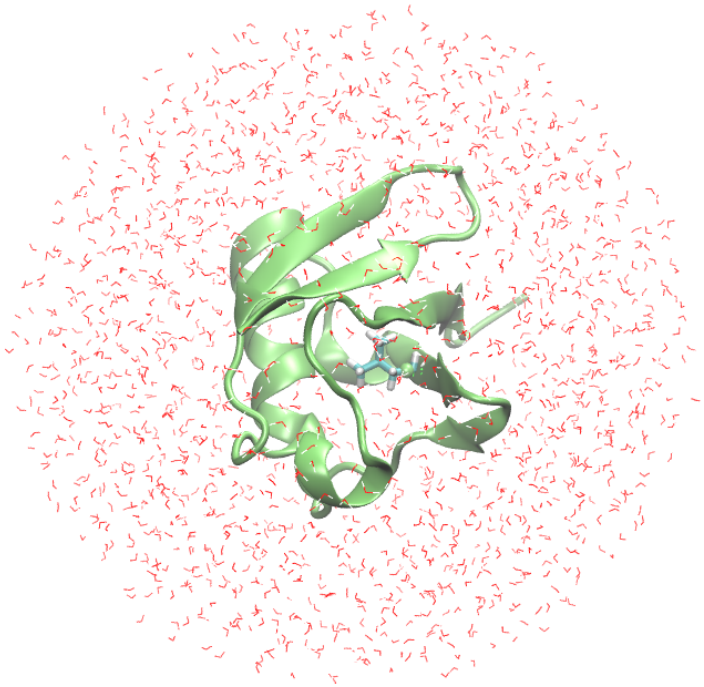
Hands-on QM/MM Tutorial

Files are also available on:
`http://villekaila.wordpress.com/`
and on the CHARMM web page

`Our files are also provided on`
`http://github.com/RowleyGroup/charmm-turbomole-examples/tree/master/qmmm`

Prof. Ville R. I. Kaila
Department of Chemistry, Technical University of Munich
E-mail: ville.kaila@ch.tum.de

- Example: QM/MM MD simulation of ubiquitin in water:



Build system using

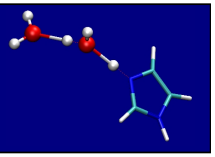
1) **VMD**
Extensions → Modeling → AutoPSF
→ Add Solvation → Add Ions
(not recommended for complex systems)

or write psfgen script
example make.tcl
followed by **solvate** and **autoionize**
(recommend for complex systems)

or

2) **CHARMM-GUI**
a nice web-based portal, which also gives you a charmm input file

Solvating the protein with VMD

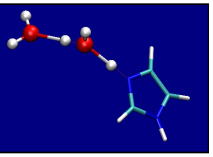


see NAMD/VMD tutorials
http://www.ks.uiuc.edu/Training/Tutorials/namd-index.html

VMD has the commands:

```
solvate protein.psf protein.pdb -t 15 -o protein-wb

autoionize -psf protein-wb.psf -pdb protein-wb.pdb -sc 0.15
```

Using CHARMM-GUI to prepare input files



www.charmm-gui.org

CHARMM-GUI

Effective Simulation Input Generator and More

CHARMM is a versatile program for atomic-level simulation o
many-particle systems, particularly macromolecules o
biological interest. - M. Karplu

about us :: input generator :: archive :: charmm docs :: lectures :: movie gallery :: video demo :: citations :: update log :: jobs & events :: givin

Some lectures and job postings are now available. See upload log for update history and giving for donation. Contact info is given below.

CHARMM-GUI

- About Us
- Input Generator
- ST-analyzer
- Archive
- CHARMM Docs
- Lectures
- Movie Gallery
- Video Demo
- Citations
- Update Log
- Jobs & Events
- Giving

Geographical Visitors

Front Page

CHARMM-GUI provides a web-based graphical user interface to generate various molecular simulation systems and input files to facilitate and standardize the usage of common and advanced simulation techniques. Currently, CHARMM-GUI supports CHARMM, NAMD, GROMACS, AMBER, GENESIS, LAMMPS, Desmond, OpenMM, and CHARMM/OpenMM simulation programs mostly based on the CHARMM force fields.

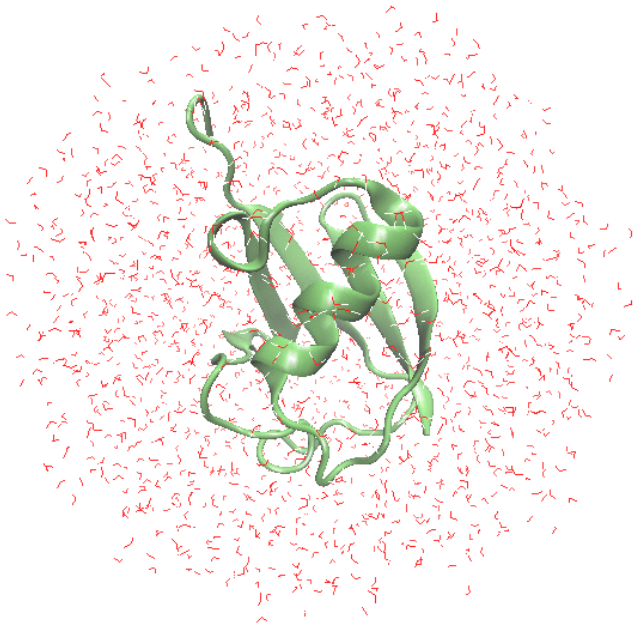
CHARMM-GUI is powered by CHARMM, an academic research program used world-wide for macromolecular dynamics and mechanics (http://www.charmm.org). Its development began in the research group of Professor Martin Karplus at Harvard University and continues throughout the world with contributing developers. CHARMM performs standard molecular dynamics and energy minimization with the potential energy functions for proteins, nucleic acids, lipids, carbohydrates, and various small molecules. In addition, CHARMM can be used for various chemical and conformational free energy calculations with many types of restraints.

The CHARMM-GUI team hopes that the tools and materials offered here are useful and helpful for your research and education.

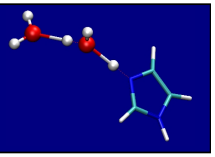
Academic users can obtain CHARMM freely from http://www.charmm.org.

MD relaxation

- Before you start with QM/MM simulations → properly relax the system by classical MD



QM/MM input for CHARMM



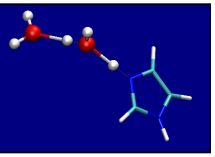
```
* QM/MM calculation of a protein, water, ion system
*

! QM/MM protein tutorial
! by Ville R. I. Kaila
! Technische Universitaet Muenchen
! Department Chemie
! E-mail: ville.kaila@ch.tum.de
! Ubiquitin in water


 ! qturboexe points to the "turbo.py" script that executes TURBOMOLE

envi qturboexe "/wrk/vkaila/DONOTREMOVE/SPRING-
SCHOOL/SPRINGSCHOOL/qmmm_protein/charmm_input/turbo711.py"
```

Charmm files start with dot -- dot

This is a comment

set variable turbo711.py python code for QM/MM

QM/MM input for CHARMM



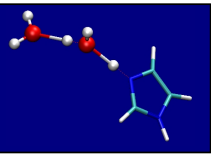
```
envi qturbooutpath "turbodir"


! "qturboinpath" includes the input files for
TURBOMOLE, e.g. control, basis, ...
envi qturboinpath "data"
```

"qturbooutpath" is the directory that includes "coord", "point_charges", and "gradient" files that are mutual between CHARMM and TURBOMOLE
(coord and point charges are updated ! before every QM calculation by CHARMM, gradient is provided by TURBOMOLE and read by CHARMM)

```
! creating the directories in the current working
directory
system "mkdir -p turbodir"
system "mkdir -p turbo_tmp"
system "mkdir -p data"
system "mkdir -p charmm_output"
```

create directories where the TURBOMOLE files are run

QM/MM input for CHARMM



```
BOMBLEV -2

set a "toppar/top_all22_prot.rtf"
set b "toppar/par_all22_prot.prm"
set c "toppar/toppar_water_ions.str"

! read in a, b, c
read rtf card name @a

read para card flex name @b

! append the nucleic acid parameters
read para card flex append name @c

 OPEN UNIT 1 READ CARD NAME "toppar/qqh.rtf"
 READ RTF CARD UNIT 1 append
 CLOSE UNIT 1

 OPEN UNIT 1 READ CARD NAME "toppar/qqh.prm"
 READ PARA CARD UNIT 1 append flex
 CLOSE UNIT 1
```

sets debugging level in CHARMM
goes to -5 (only for testing stuff!)

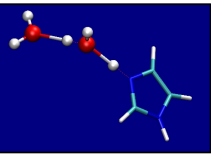
read in topology & parameter files for MM

a bit charmming by creating variables a, b, c and then reading them

the flex command is used for appending

here we read in parameters for the link atoms
(we need dummy parameters with "zeros")

Lets have a look at the CHARMM topology file



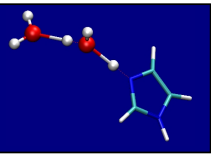
```
*>>>>>>>>>CHARMM22 All-Hydrogen Topology File for Proteins <<<<<<
*>>>>> Includes phi, psi cross term map (CMAP) correction <<<<<<<
*>>>>>>>>>>>>>>>>>>>>>> December, 2003 <<<<<<<<<<<<<<<<<<<<<<<<
* All comments to ADM jr. via the CHARMM web site: www.charmm.org
*               parameter set discussion forum
*
31  1
```

topology file starts with dot dots

```
MASS    41 H      1.00800 H ! polar H
MASS    42 HC     1.00800 H ! N-ter H
```

and then mass of all elements

Lets have a look at the CHARMM topology file



resname

total charge/partial charges

```
RESI ALA          0.00
GROUP
ATOM N    NH1    -0.47  !      |
ATOM HN   H       0.31  !   HN-N
ATOM CA   CT1     0.07  !      |      HB1
ATOM HA   HB      0.09  !      |     /
GROUP                   !   HA-CA--CB-HB2
ATOM CB   CT3    -0.27  !      |     \
ATOM HB1  HA      0.09  !      |      HB3
ATOM HB2  HA      0.09  !    O=C
ATOM HB3  HA      0.09  !      |
GROUP                   !
ATOM C    C       0.51
ATOM O    O      -0.51
```

atom name
(as in **pdb** file)

atom type
(refers to the parameter file)
atom types & charges
are written to the **psf** file

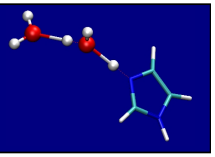
create bond between CB and CA
create bond between N HN
etc.

CHARMM
backbone correction CMAP

```
BOND CB CA  N  HN  N  CA
BOND C  CA  C  +N  CA HA  CB HB1  CB HB2  CB HB3
DOUBLE O  C
IMPR N -C CA HN  C CA +N O
CMAP -C  N  CA  C   N  CA  C  +N
DONOR HN N
ACCEPTOR O C
IC -C   CA   *N   HN    1.3551 126.4900  180.0000 115.4200  0.999
IC -C   N    CA   C     1.3551 126.4900  180.0000 114.4400  1.539
....
```

internal coordinate list
(helps to build the residue)

Lets have a look at the CHARMM parameter file (1/3)



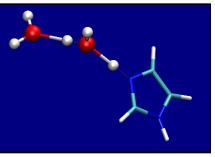
```
BONDS
!V(bond) = Kb(b - b0)**2
!Kb: kcal/mole/A**2
!b0: A
!atom type Kb         b0

CA    CA     305.000      1.3750
...
```

force constant and eq. distance for bonding term between atom types CA and CA

```
ANGLES
!V(angle) = Ktheta(Theta - Theta0)**2
!Ktheta: kcal/mole/rad**2
!Theta0: degrees

CT1   CT1   C       52.000    108.0000
...
```

force constant and eq. distance for angle terms between atom types CT1-CT1-C

Lets have a look at the CHARMM parameter file (2/3)

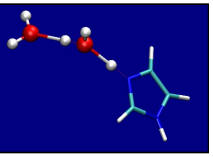


```
DIHEDRALS
!
!V(dihedral) = Kchi(1 + cos(n(chi) - delta))
!
!Kchi: kcal/mole
!n: multiplicity
!delta: degrees
!
!atom types             Kchi    n   delta
!

CE2  CE1  CT2  HA       0.1200  3     0.00 !

IMPROPER
!
!V(improper) = Kpsi(psi - psi0)**2
!
!Kpsi: kcal/mole/rad**2
!psi0: degrees
!note that the second column of numbers (0) is ignored
!
!atom types           Kpsi                   psi0
O    X    X    C     120.0000          0      0.0000
```

force constant and eq. dihedrals
for dihedral terms
between atom types
CE2-CE1-CT2-HA

atom types
O-X-X-C kept at 120 deg
planarity

Lets have a look at the CHARMM parameter file (3/3)

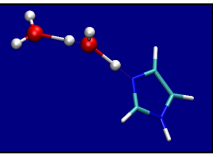


```
!V(Lennard-Jones) = Eps,i,j[(Rmin,i,j/ri,j)**12 - 2(Rmin,i,j/ri,j)**6]
!
!epsilon: kcal/mole, Eps,i,j = sqrt(eps,i * eps,j)
!Rmin/2: A, Rmin,i,j = Rmin/2,i + Rmin/2,j
!
!atom  ignored    epsilon      Rmin/2   ignored   eps,1-4       Rmin/2,1-4

SOD      0.0         -0.0469     1.41075
```

Lennard-Jones terms for atom types "SOD"

... lets now continue with input file

QM/MM input contd.



```
open read card unit 10 name "segments/test_proa.pdb"
read sequence pdb unit 10
generate PROA setup warn first NTER last CTER

read coor pdb  unit 10 resid

! read in water segment WATA
open read card unit 10 name "segments/test_wata.pdb"
read sequence pdb unit 10
generate WATA setup warn noangle nodihedral

read coor pdb  unit 10 resid

!open read card unit 10 name "./sod.pdb"
!read sequence pdb unit 10
!generate IO1 setup warn noangle nodihedral
```

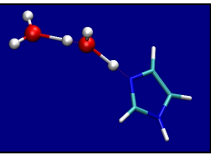
read in protein segment PROA
create NTER and CTER patch
(look at PRES NTER/CTER in topology file)

read in water segment WATA
.. in charmm you need to have different segements for protein subunits, water, ions, etc.

YES IT IS LIKE FORTRAN...

read in ions
→ not in this example
(we commented them out)

QM/MM input contd.



```
! print out pdb structure
open write card unit 10 name "charmm_output/test_all.pdb"
write coor pdb  unit 10

! start from the  coordinates of
! a previous run
!open read unit 22 card name "charmm_input/mini_all.crd"
!read coor unit 22 card


!OPEN UNIT 31 READ CARD
NAME  "charmm_input/structure.psf"
!READ PSF CARD UNIT 31
!CLOSE UNIT 31

!OPEN UNIT 31 READ CARD
NAME  "charmm_input/structure.crd"
!READ COOR CARD UNIT 31
!CLOSE UNIT 31
```

write combined coordinates into pdb , crd,

if you already have a crd and psf file you can read in them instead of creating your system

NOTE: charmm has a different psf format than NAMD, which uses xplor format

QM/MM input contd.

```
ENERGY
```

... now finally starts QM/MM specific part

classical single point energy

```
addlinkatom QQH1 PROA 43 CB  PROA 43 CA
```

order qm – mm

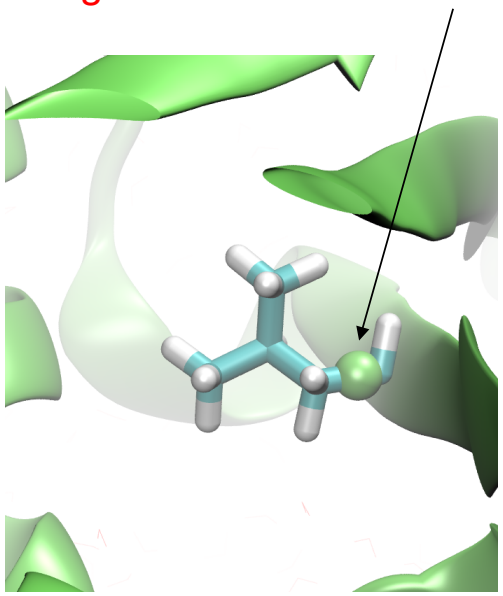
create a link atom "QQH1" between CB and CA of resid 43 in segname PROA

```
define qm sele –
atom PROA 43 CB  –
.or. atom PROA 43 HB1 –
.or. atom PROA 43 HB2 –
.or. atom PROA 43 CG  –
.or. atom PROA 43 HG  –
.or. atom PROA 43 CD1 –
.or. atom PROA 43 HD11 –
.or. atom PROA 43 HD12 –
.or. atom PROA 43 HD13 –
.or. atom PROA 43 CD2 –
.or. atom PROA 43 HD21 –
.or. atom PROA 43 HD22 –
.or. atom PROA 43 HD23 –
show end
```

line continues:

define the qm system

NOTE:
There is no physical/chemical purpose of simulating resid 43 by QM/MM (example is only for demonstration purposes)



QM/MM input contd.

```
lonepair colinear scaled -.7261 -
sele type QQH1 show end sele atom PROA 43 CB  -
show end sele atom PROA 43 CA show end


print coord sele qm .or. atom * * QQH* end

stop
```

execute:

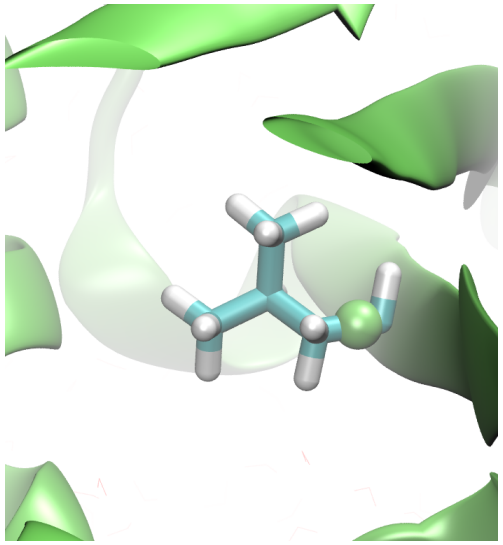
```
./charmm < qmmm_protein.inp > qmmm_protein.out
```

in order to place the link atom in the correct direction use lone pair command

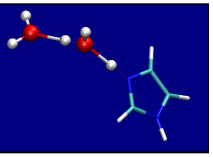
print the qm region and stop the code

→

take coordinates and perform TURBOMOLE single point

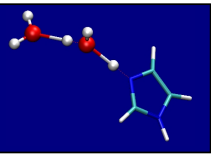


TURBOMOLE single point for QM/MM (1/5)



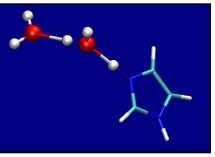
```
cd data
```
prepare coord.xyz based on what charmm prints out that contains the qm region
```
x2t coor.xyz > coord        for this use need: module load turbomole

define
IF YOU WANT TO READ DEFAULT-DATA FROM ANOTHER control-TYPE FILE,
THEN ENTER ITS LOCATION/NAME OR OTHERWISE HIT >return<.
hit enter
INPUT TITLE OR                                                    how to run define
ENTER & TO REPEAT DEFINITION OF DEFAULT INPUT FILE
hit enter
IF YOU APPEND A QUESTION MARK TO ANY COMMAND AN EXPLANATION
OF THAT COMMAND MAY BE GIVEN
a coord
IF YOU APPEND A QUESTION MARK TO ANY COMMAND AN EXPLANATION
OF THAT COMMAND MAY BE GIVEN
*
IF YOU DO NOT WANT TO USE INTERNAL COORDINATES ENTER no
no ! internal coordinates will mess up the qm system positioni tioning
relative to mm point charges
```

TURBOMOLE single point for QM/MM (2/5)



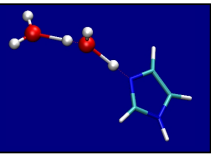
```
ATOMIC ATTRIBUTE DEFINITION MENU ( #atoms=198 #bas=198 #ecp=2 )
b

ENTER A SET OF ATOMS TO WHICH YOU WANT TO ASSIGN BASIS SETS
( ATOMIC SET : all none )
TO OUTPUT ATOMIC SET SYNTAX ENTER A QUESTION MARK ?
E.G. all dz (DZ BASIS FOR ALL ATOMS)
all sto-3g hondo (SCALED STO-3G)
1,2,4-6 dzp (DZP BASIS FOR ATOMS 1,2,4,5,6)
“c” tz (TZ BASIS FOR ALL CARBON ATOMS)
ANY DISPLAY COMMAND dis MAY BE ENTERED OR YOU MAY
HIT >return< TO QUIT (GOING BACK TO MAIN MENU)

all def2-SVP
*
```

TURBOMOLE single point for QM/MM (3/5)



```
CHOOSE COMMAND
infsao : OUTPUT SAO INFORMATION
atb : Switch for writing MOs in ASCII or binary format
eht : PROVIDE MOS && OCCUPATION NUMBERS FROM EXTENDED HUECKEL GUESS
use : SUPPLY MO INFORMATION USING DATA FROM
man : MANUAL SPECIFICATION OF OCCUPATION NUMBERS
hcore : HAMILTON CORE GUESS FOR MOS
flip : FLIP SPIN OF A SELECTED ATOM
& : MOVE BACK TO THE ATOMIC ATTRIBUTES MENU
THE COMMANDS use OR eht OR * OR q(uit) TERMINATE THIS MENU !!!
FOR EXPLANATIONS APPEND A QUESTION MARK (?) TO ANY COMMAND

eht
ENTER THE MOLECULAR CHARGE (DEFAULT=0)
0
! put in here the total charge of you qm region
DO YOU ACCEPT THIS OCCUPATION ? DEFAULT=y
hit enter
```

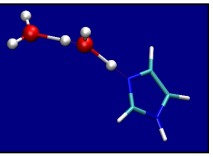
TURBOMOLE single point for QM/MM (4/5)



```
GENERAL MENU : SELECT YOUR TOPIC
scf : SELECT NON-DEFAULT SCF PARAMETER
mp2 : OPTIONS AND DATA GROUPS FOR rimp2 and mpgrad
…

dft
STATUS OF DFT_OPTIONS:
DFT is NOT used
functional b-p
gridsize m3
ENTER DFT-OPTION TO BE MODIFIED
func : TO CHANGE TYPE OF FUNCTIONAL
grid : TO CHANGE GRIDSIZE
on: TO SWITCH ON DFT
Just , q or ‘*’ terminate this menu.
on
func b3-lyp
*
```

```
dsp
on
*

terminate define
*
```

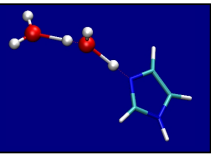
TURBOMOLE single point for QM/MM (5/5)



```
in control file change:
$scfiterlimit 500
$thize 0.10000000E+99
$scfdamp start=1.700 step=0.050 min=0.050

remove line: $scfdump
(can also be done with "kdg")

run a scf iteration

dscf > dscf.out

in control file $drvopt section add:
point charges
at the end of the control file add:
$point_charges file=point_charges
$point_charge_gradients file=pc_gradient
```

this will calculate a QM force on MM atoms, add MM charges around QM system, and calculate the gradients form MM on QM

Perform QM/MM single point energy evaluation



```
!stop
```

comment this out

scale the mass of ink atoms to 1.008

```
 scalar mass show select type qq* show end
 scalar mass set  1.008000 select type qq* show end
 scalar mass show select type qq* show end
```

```
 NBOND -
     atom   cdiel  eps 1.0 -                ! Electrostatics
     vatom  vswitch        -                ! VdW
     elec   switch         -                ! Elec
     ctonnb 100.0 ctofnb 120.0  cutnb 140.0
```

define cutoffs for electrostatics and vdw
no PME yet for DFT/MM

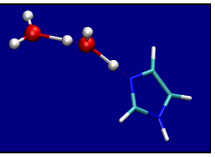
```
qturbo remove sele qm end
```

call turbomole
remove all MM contributions for QM definition

```
 ENERGY
```

evaluate QM/MM energy

Perform QM/MM single point energy evaluation



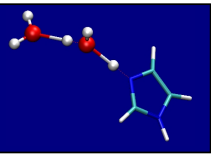
```
open write unit 41 file name "optimize-final.dcd"

mini abnr nstep 500 nprint 1 tolg 0.0006 -
tole 0.0006 IUNCrd 41 NSAVC 1

write coor pdb  unit 10
* minimized
*

open write card unit 10 name "minimized.crd"
write coor unit 10 card
* minimized
*

open write unit 10 card name "minimized.psf"
write psf  unit 10 card
* minimized
*

stop
```

open trajectory file

perform geometry optimization with Adopted Basis Newton-Raphson for 500 se or until tolerance in energy/gradient is less than 0.0006 (kcal/mol)

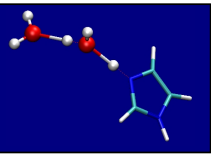
write coordinates and psf file

Perform QM/MM MD



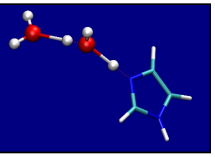
```
OPEN UNIT 44 write   FORMatted NAME "charmm_output/dynamics.res"
OPEN UNIT 42 write UNFORMatted NAME "charmm_output/dynamics.dcd"
open write unit 31 card name "charmm_output/dynamics.dat"
open unit 20 write unform name "charmm_output/velocities.trj"

set STEP 500
SCALAR FBETA SET 8.0 select all end

dynamics lang leap start -
        timestep 0.001       - ! timestep in picoseconds
        nstep @STEP          - ! number of steps and energy evaluations
        nprint 1             - ! step frequency for writing in kunit and pr energy on unit 6
        iunwri 44            - ! unit to write restart file
        iunrea -1            - ! unit to read restart file
        iuncrd 42            - ! unit to write coordinates (unformatted)
        iunvel 20            - ! unit to write velocities out to
        kunit 31             - ! unit to write energy and temp inforamtion out to file
        iprfrq 20             - ! frequency for avg and rms energy
        firstt 310.          - ! initial T for velocity assignments
        tstruc 310.          - ! initial T for velocity assignments
        finalt 310.          - ! final T
        tbath 310.           - ! temperature of heat bath
        iasors 1             - ! asgn (NOT scale) veldur heat/equil (.ne. 0
```

restarting QM/MM MD (1/2)



```
OPEN UNIT 11 read    FORMatted NAME "charmm_output/dynamics.res"
OPEN UNIT 44 write   FORMatted NAME "charmm_output/dynamics2.res"
OPEN UNIT 42 write UNFORMatted NAME "charmm_output/dynamics2.dcd"
open write unit 31 card name "charmm_output/dynamics2.dat"
open unit 20 write unform name "charmm_output/dynamics2.trj"

set STEP 500
SCALAR FBETA SET 8.0 select all end
```

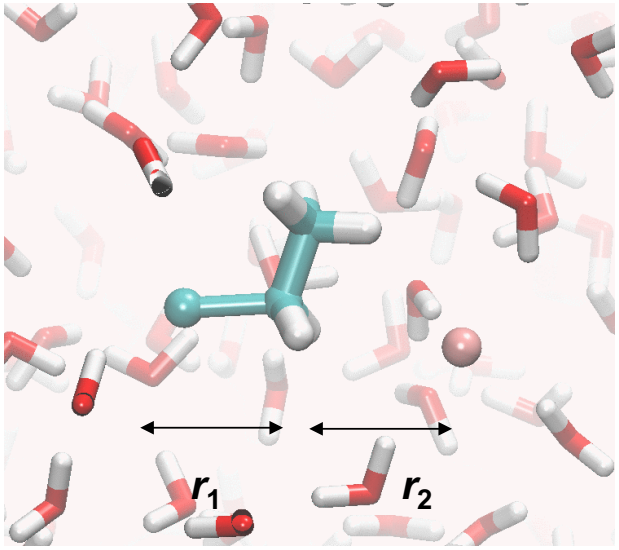
restarting QM/MM MD (2/2)



```
dynamics lang leap RESTRT -
         timestep 0.001       - ! timestep in picoseconds
         nstep @STEP          - ! number of steps and energy evaluations
         nprint 1             - ! step frequency for writing in kunit and pr energy on unit 6
         iunwri 44            - ! unit to write restart file
         iunrea 11            - ! unit to read restart file
         iuncrd 42            - ! unit to write coordinates (unformatted)
         iunvel 20            - ! unit to write velocities out to
         kunit 31             - ! unit to write energy and temp inforamtion out to file
         iprfrq 20             - ! frequency for avg and rms energy
         firstt 310.          - ! initial T for velocity assignments
         tstruc 310.          - ! initial T for velocity assignments
         finalt 310.          - ! final T
         tbath 310.           - ! temperature of heat bath
         iasors 1             - ! asgn (NOT scale) veldur heat/equil (.ne. 0 = asgn; .eq. 0 = scale)
         iasvel 1             - ! use gaussian distrib of vel (.gt. 0 =gaussian ; .lt. 0 = uniform)
         nsavc 1              - ! freq for writing coordinates
         nsavv 1              - ! freq for writing velocities
         isvfrq 100           - ! frequency for writing restart files
      ihtfrq 0            - ! frequency for heating steps during the dynamics
      ieqfrq 0            - ! frequency for scaling velocities during heating
      inbfrq 1            - ! lists updated when necessary (heuristic test)
      ihbfrq 0            - ! frequency for updating hydrogen bond list
      ntrfrq 0            - ! step freq for stopping rotation and translation
      echeck 100000000.0  - ! max variation of energy step-to-step
      ichecw 0              ! check to see if avg. temp. lies w/temp range (0 = do not check)

OPEN UNIT 1 WRITE CARD NAME "charmm_output/dynamics2.pdb"
WRITE COOR PDB UNIT 1
```

Example 2: Free energy surface of chemical reaction in solution

- QM/MM free energy simulation
  $Cl^- + CH_3CH_2\text{-}Br \rightarrow Cl\text{-}CH_3CH_2 + Br^-$



reaction coordinate:
$\boldsymbol{R}=\boldsymbol{r}_1-\boldsymbol{r}_2$

employ bias on the reaction coordinate:

$V_b = ½\ k\ (\boldsymbol{R}-r_0)^2$

run 1 → r_0 = -2.0
(reactant: Cl-C is short, C...Br is long)

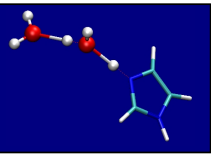
...

run 15 → r_0 = +1.7
(product: Cl...C is long, C-Br is short)

compute unbiased prob. distr. p(R) based on biased data $p_b(\boldsymbol{R})$ and V_b

unbiased potential of mean force for $\boldsymbol{R}$ → $G(\boldsymbol{R})$

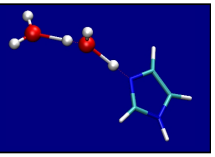
How to set this up in practice?



- Introduce restraints using the RESD module in CHARMM

```
SET ATOM1 PROA 1 CL
SET ATOM2 PROA 1 C1
SET ATOM3 PROB 1 BR

! 1.85-3.49 = -1.64
SET V -1.64

SKIP NONE
RESDistance  RESET KVAL 100.0  RVAL @v -
1.0 @atom1 @atom2 -1.0 @atom2 @atom3
```

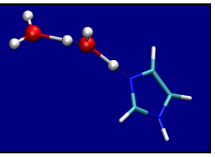
How to set this up in practice?



- start by a short minimization that relaxes the restrained coordinate (as in example 1)

```
mini abnr nstep 10 nprint 1 tolg 0.0001 tole 0.0001 IUNCrd 41 NSAVC 1

open write card unit 10 name "mini.pdb"
write coor pdb  unit 10
* minimized
*

open write card unit 10 name "mini.crd"
write coor unit 10 card
* minimized
*

open write unit 10 card name "mini.psf"
write psf unit 10 card
* minimized
*
```

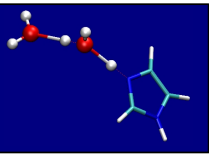
How to set this up in practice?



- ... then run restrained dynamics

```
open write unit 42 file name "./trajectory.dcd"
open unit 44 write card name "./dyn.res"
open write unit 31 card name "./production.dat"
open unit 20 write unform name "./velocities3.trj"

! for 10000 x 1 fs steps
set STEP 10000

! coupling for langevin dynamics
SCALAR FBETA SET 8.0 select all end
```

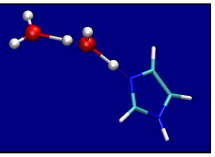
How to set this up in practice?



- ... then run restrained dynamics

```
dynamics lang leap start -
       timestep 0.001        - ! timestep in picoseconds
       nstep @STEP           - ! number of steps and energy evaluations
       nprint 1              - ! step frequency for writing in kunit and pr energy on unit 6
       iunwri 44             - ! unit to write restart file
      iunrea -1          - ! unit to read restart file
       iuncrd 42             - ! unit to write coordinates (unformatted)
       iunvel 20             - ! unit to write velocities out to
      kunit 31           - ! unit to write energy and temp inforamtion out to file
      iprfrq 20           - ! frequency for avg and rms energy
      firstt 298.        - ! initial T for velocity assignments
      tstruc 298.        - ! initial T for velocity assignments
       finalt 298.           - ! final T
      tbath 298.         - ! temperature of heat bath
      iasors 1           - ! asgn (NOT scale) veldur heat/equil (.ne. 0 = asgn; .eq. 0 = scale)
      iasvel 1           - ! use gaussian distrib of vel (.gt. 0 =gaussian ; .lt. 0 = uniform)
      nsavc 1            - ! freq for writing coordinates
      nsavv 1            - ! freq for writing velocities
      isvfrq 100         - ! frequency for writing restart files
      ihtfrq 0           - ! frequency for heating steps during the dynamics
      ieqfrq 0           - ! frequency for scaling velocities during heating
      inbfrq 1           - ! lists updated when necessary (heuristic test)
      ihbfrq 0           - ! frequency for updating hydrogen bond list
      ntrfrq 0           - ! step freq for stopping rotation and translation
      echeck 100000000.0 - ! max variation of energy step-to-step
      ichecw 0             ! check to see if avg. temp. lies w/temp range (0 = do not check)
```

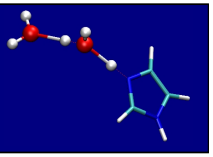
How to set this up in practice?



- ... and write final coordinates:

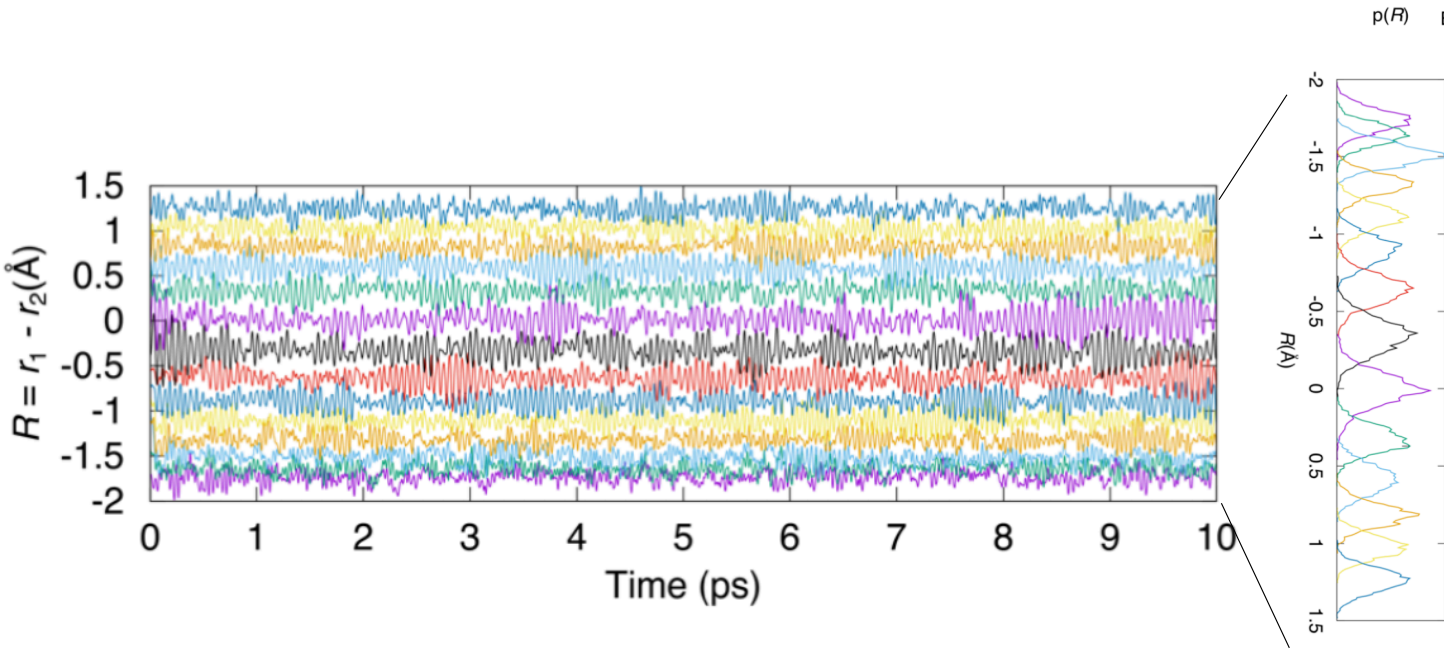
```
open write card unit 10 name "dynamics_all.pdb"
write coor pdb  unit 10
* dynamics
*

open write card unit 10 name "dynamics_all.crd"
write coor unit 10 card
* dynamics
*

open write unit 10 card name "dynamics_all.psf"
write psf  unit 10 card
* dynamics
*

stop
```

How to set this up in practice?

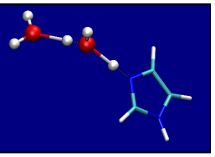


- extract distances from restrained runs

```
us_0.dat   us_11.dat  us_13.dat  us_15.dat  us_2.dat  us_4.dat us_6.dat  us_8.dat
us_10.dat  us_12.dat  us_14.dat  us_1.dat   us_3.dat  us_5.dat us_7.dat  us_9.dat
```



How to set this up in practice?



- wham together

http://membrane.urmc.rochester.edu/content/wham

```
$ ./wham
Command line: wham [P|Ppi|Pval] hist_min hist_max num_bins tol temperature numpad
metadatafile freefile [num_MC_trials randSeed]
```

```
./wham  -2 1.5 16 0.000001 310 0 metadata.dat freefile
```

leave out P if not periodic

- plot your data

```
gnuplot
p 'freefile' u 1:2
```

Analyze your PMF