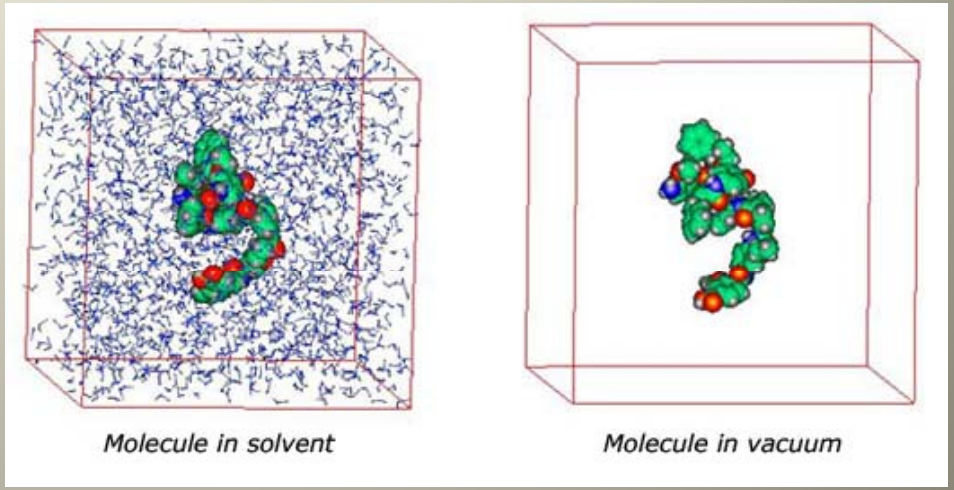
GROMACS

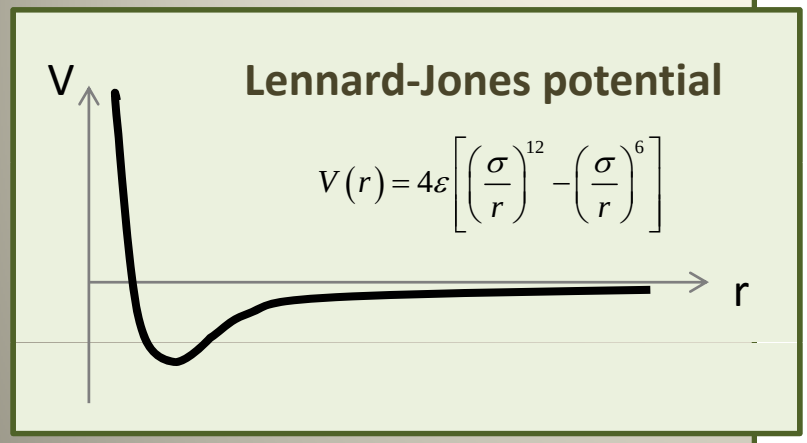
Groningen Machine for Chemical Simulations

…and molecular dynamics



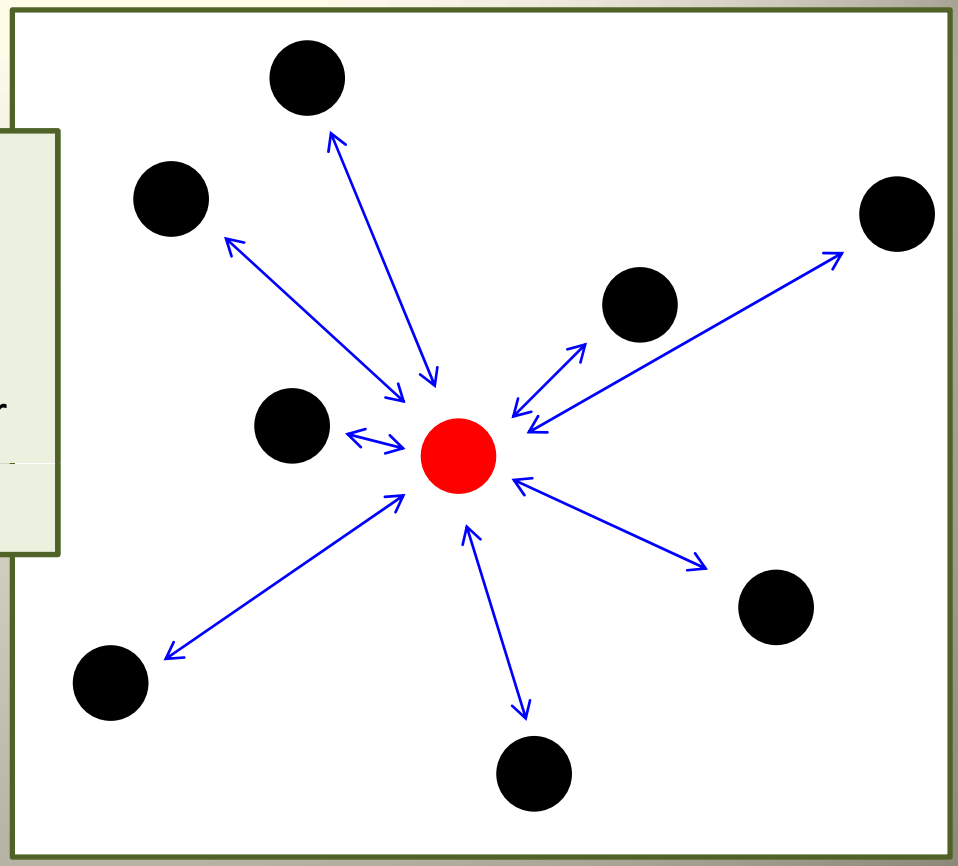
polymedix.com

Motion of particles



Lennard-Jones potential

$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

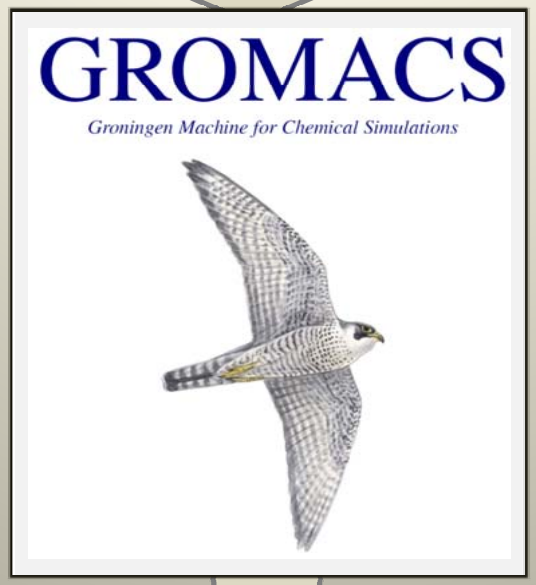


$$m_i \ddot{r}_i = f_i \qquad f_i = -\frac{\partial}{\partial r_i} V$$

With a Newtonian view, we can calculate the future of the system.

Simple MD program

Molecular structure & topology

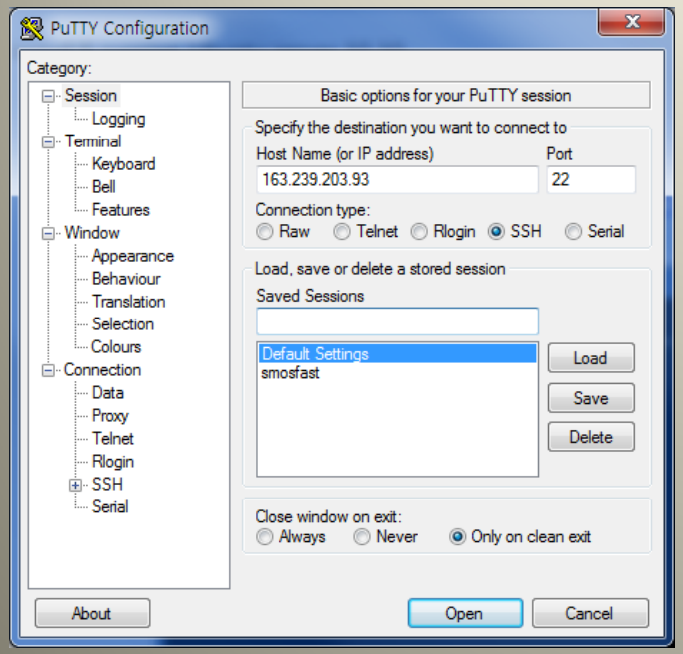


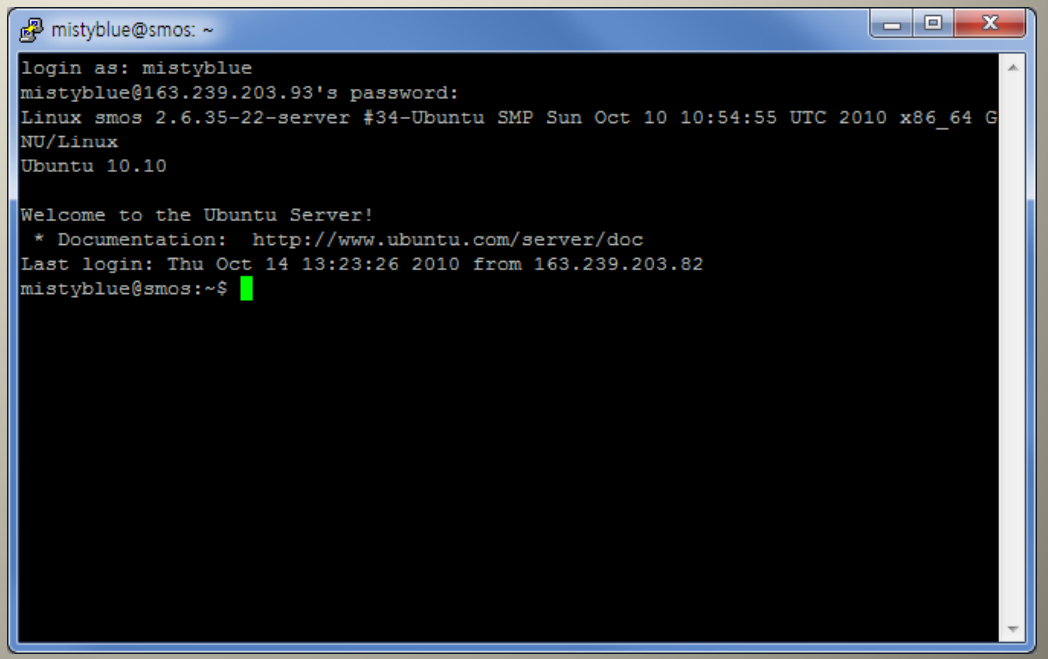
Structure & trajectory

Install & running Ubuntu server 10.10

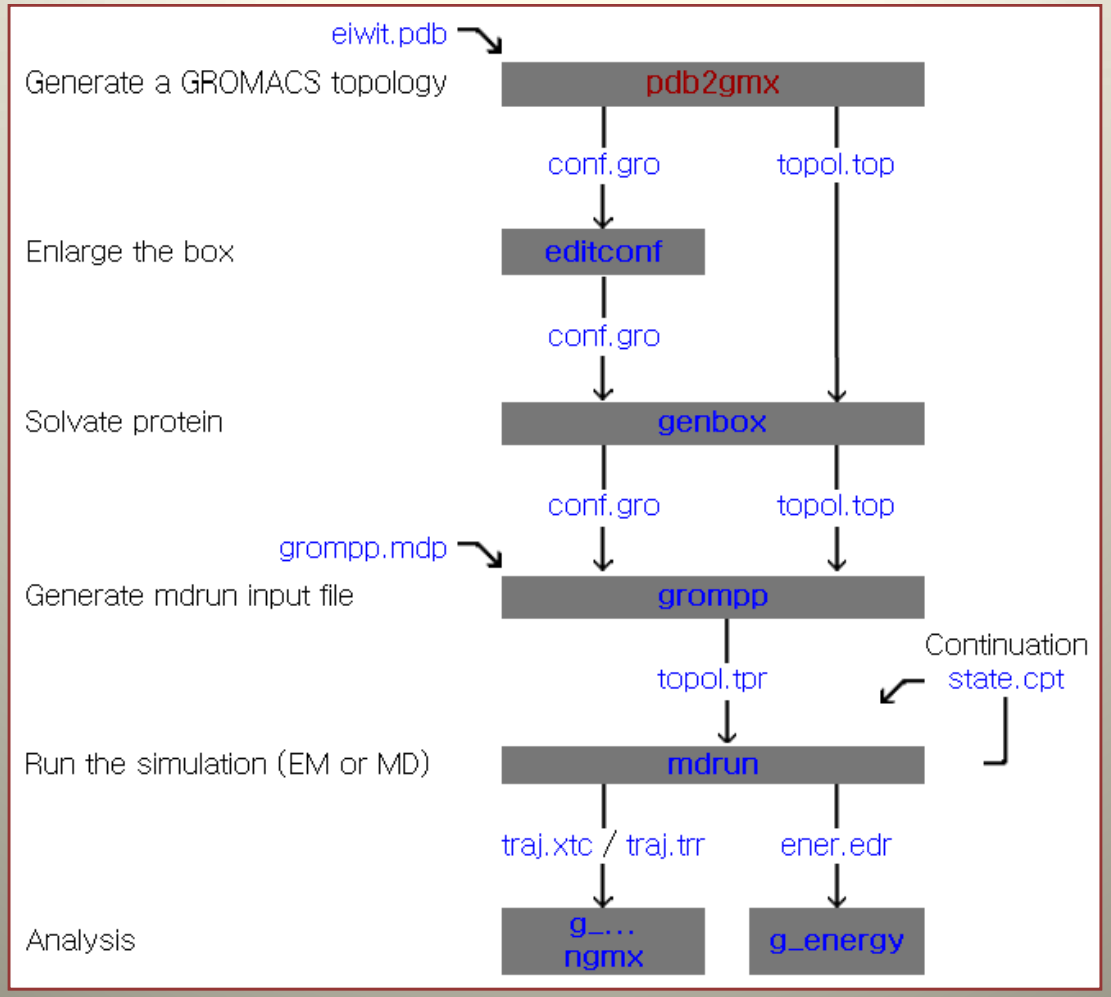
IP : 163.239.203.93

Any ssh program can be used to access to the server.
I recommend PuTTY. (http://www.chiark.greenend.org.uk/~sgtatham/putty/)
For GUI purpose, use Xming (http://www.straightrunning.com/XmingNotes/)





Flow Chart


eiwit.pdb
Generate a GROMACS topology
pdb2gmx
conf.gro
topol.top
Enlarge the box
editconf
conf.gro
Solvate protein
genbox
conf.gro
topol.top
grompp.mdp
Generate mdrun input file
grompp
Continuation
topol.tpr
state.cpt
Run the simulation (EM or MD)
mdrun
traj.xtc / traj.trr
ener.edr
Analysis
g_...
ngmx
g_energy

GROMACS files

Molecular Structure file (.gro, .pdb)

generated by the program pdb2gmx.

Molecular Topology file (.top)

generated by the program pdb2gmx.
contains a complete description of all the interactions in your peptide or protein.

Molecular Dynamics parameter file (.mdp)

contains all information about the Molecular Dynamics simulation itself e.g. time-step, number of steps, temperature, pressure etc.

Index file (.ndx)

Sometimes you may need an index file to specify actions on groups of atoms (e.g. Temperature coupling, accelerations, freezing).

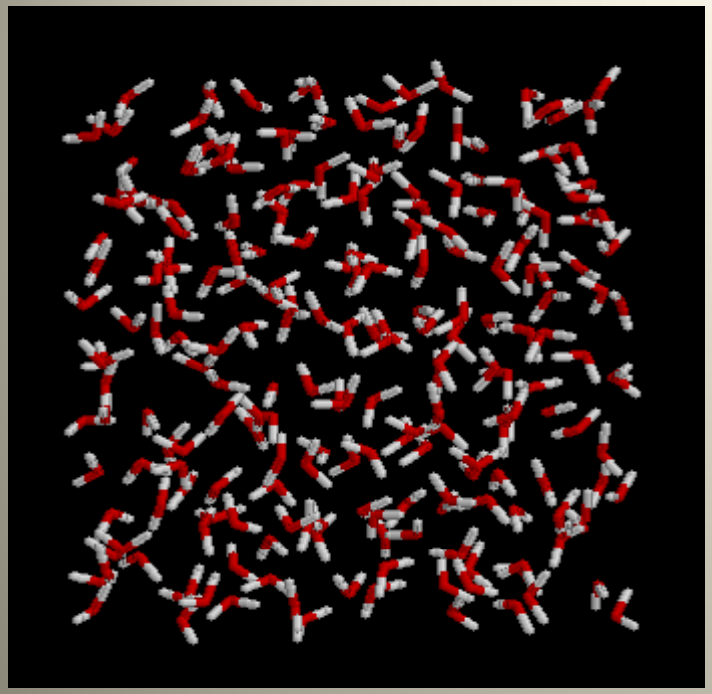
Run input file (.tpr)

to combine the molecular structure (.gro file), topology (.top file) MD-parameters (.mdp file) and (optionally) the index file (ndx) to generate a run input file.
Generated by the grompp program.

Trajectory file (.trr)

The output files of mdrun are the trajectory file and a logfile (.log file).

Water



Start with
→ spc216.pdb

Use pdb2gmx program
→ conf.gro, topol.top

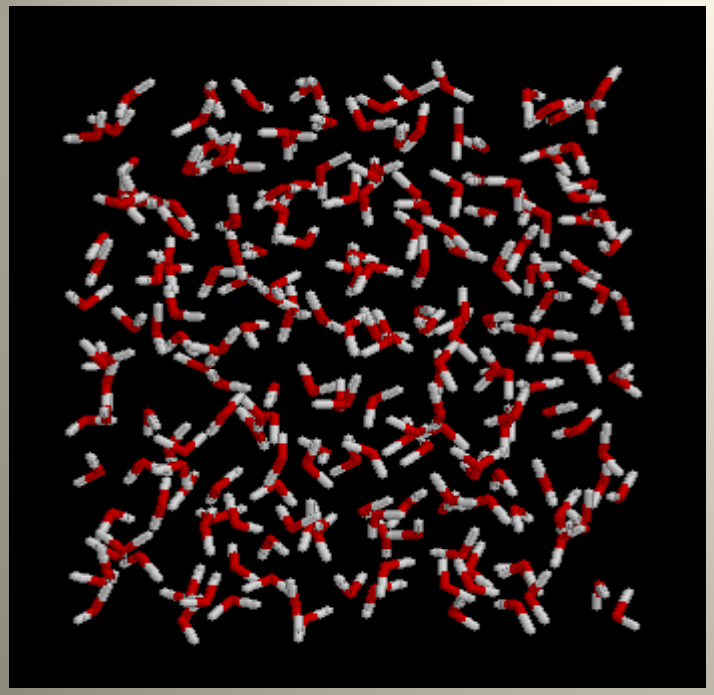
Setting the parameters in grompp.mdp

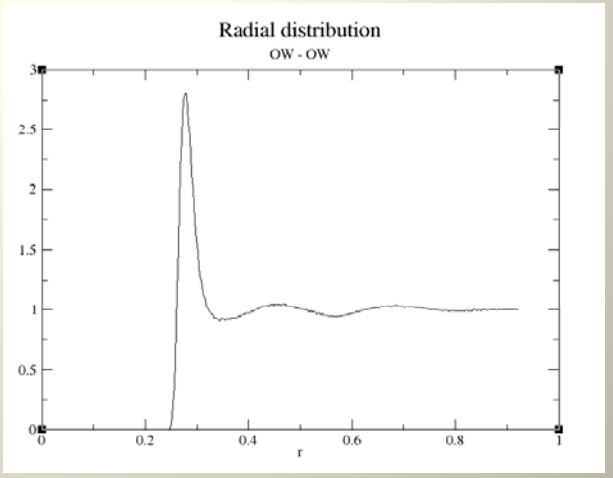
Use grompp program to generate
→ tolo.tpr

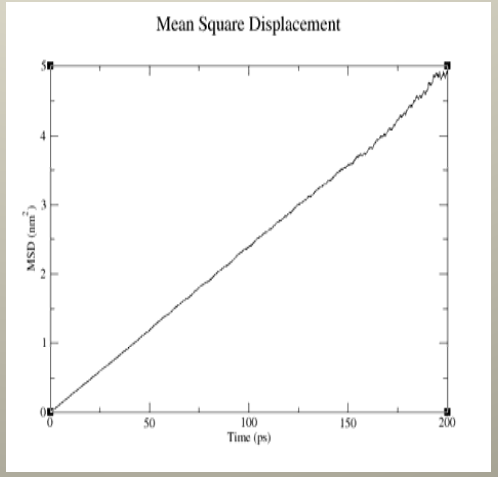
Run mdrun program

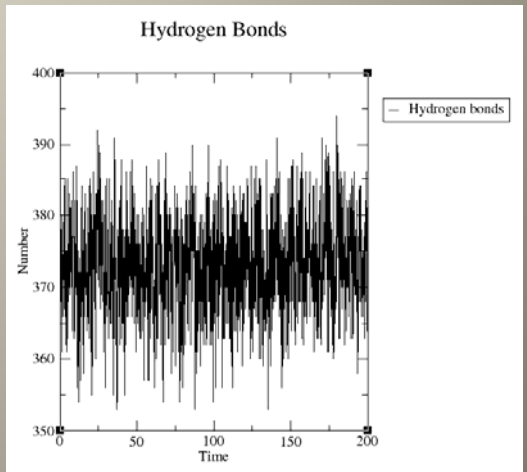
```
Rasmol spc216.pdb
more spc216.pdb

pdb2gmx –f spc216.pdb

more conf.gro
more topol.top

grompp –v –maxwarn 1

mdrun -v

ngmx
g_rdf –n index
xmgrace rdf.xvg
g_hbond
xmgrace hbnum.xvg
g_msd –n index
xmgrace msd.xvg
```

Water

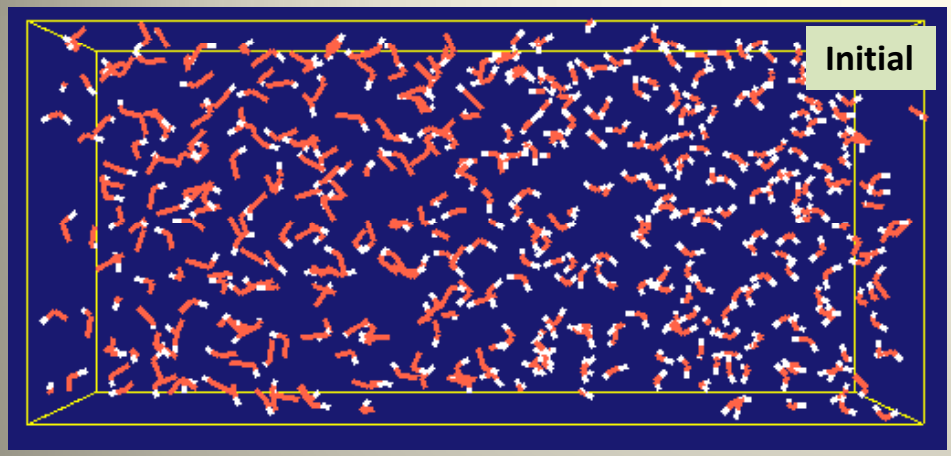


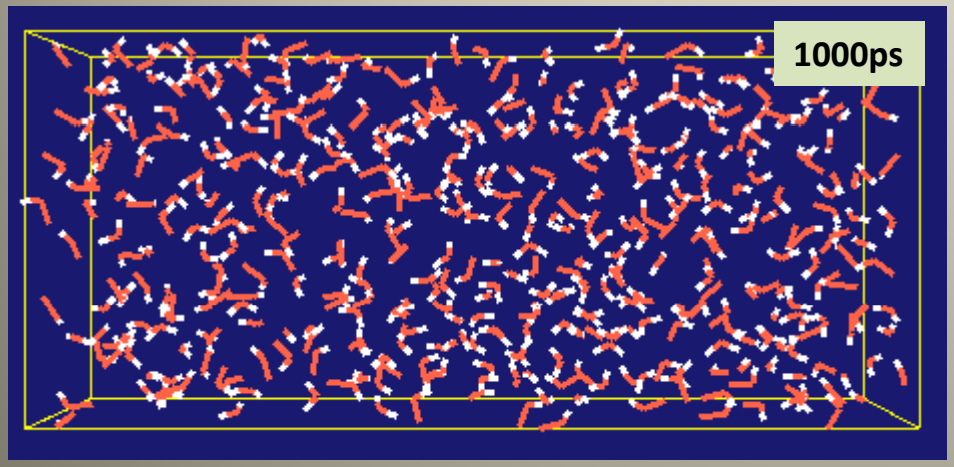

Radial distribution
OW - OW
g(r)
r
0
0.5
1
1.5
2
2.5
3
0.2
0.4
0.6
0.8

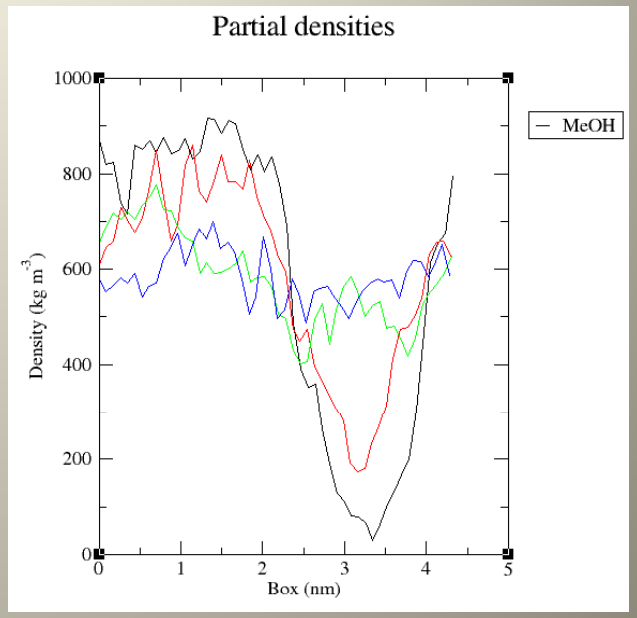

Mean Square Displacement
MSD (nm²)
Time (ps)
0
1
2
3
4
5
50
100
150
200


Hydrogen Bonds
Number
Time
Hydrogen bonds
350
360
370
380
390
400
50
100
150
200

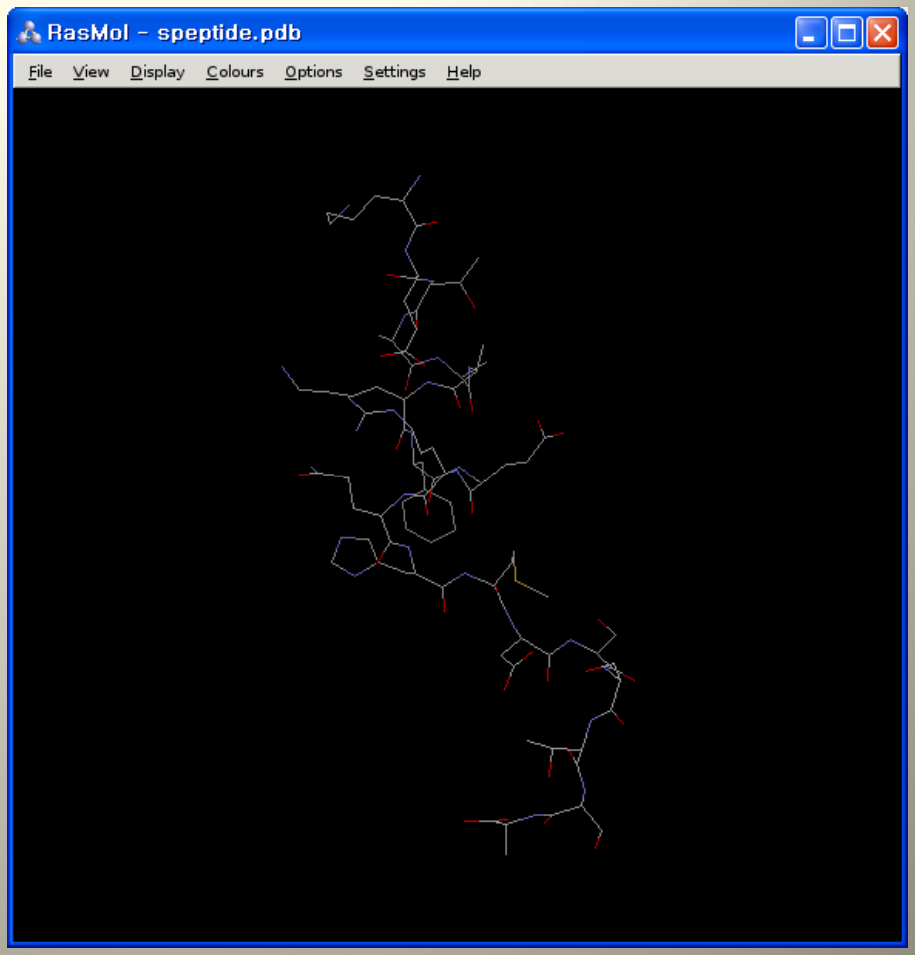
Water-methanol mixture





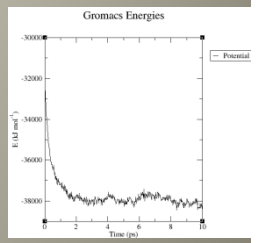


Ribonuclease S-peptide
(residues 1-20 among 124)



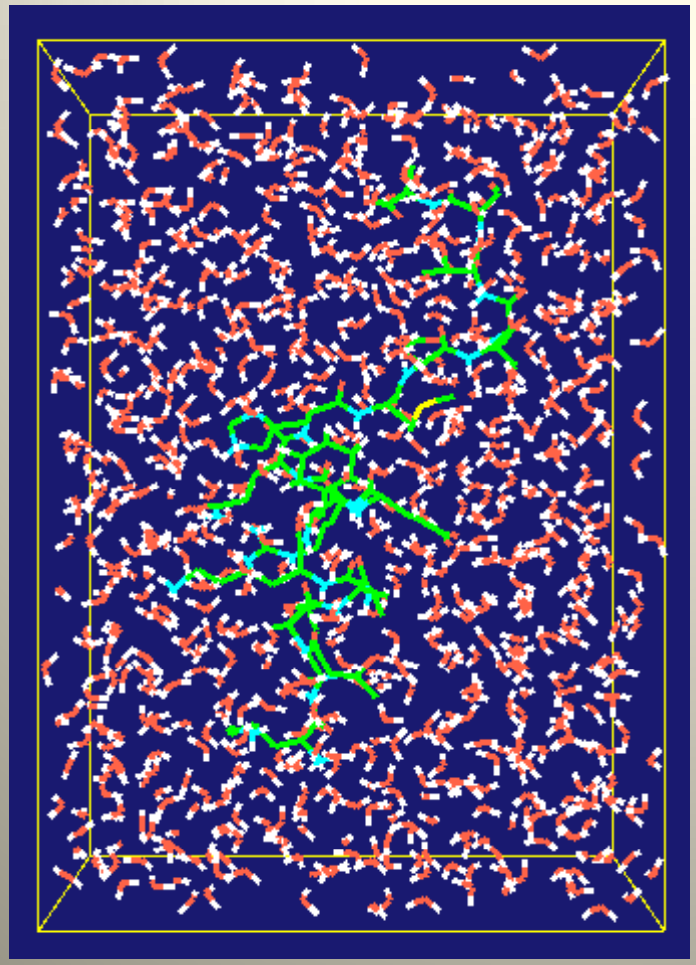
Start with em.mdp full.mdp pr.mdp **speptide.pdb**

1. **.pdb ➔ .gro and . top (pdb2gmx)**
   $ pdb2gmx -f speptide.pdb -p speptide.top -o speptide.gro → **posre.itp** **speptide.gro** **speptide.top**
2. **expand box size**
   $ editconf –f speptide –o –d 0.5 → **out.gro**
3. **solvate the peptide w. water (genbox)**
   $ genbox –cp out –cs –p speptide –o b4em → **b4em.gro** **speptide.top (modified)**
4. **generate binary input files for mdrun**
   $ grompp -v -f em -c b4em -o em -p speptide → **em.tpr** **mdout.mdp**
5. **energy minimization**
   $ mdrun -v -s em -o em -c after_em -g emlog → **after_em.gro** **emlog.log** **ener.edr** **em.trr**
6. **mdrun w. position restraints**
   $ grompp –f pr –o pr –c after_em –r after_em –p speptide → **pr.tpr** **mdout.mdp (modified)**
   $ mdrun –v –s pr –e pr –o pr –c after_pr –g prlog >& pr.job & → **after_pr.gro** **pr.edr** **pr.job** **pr.trr** **prlog.log** **state.cpt**

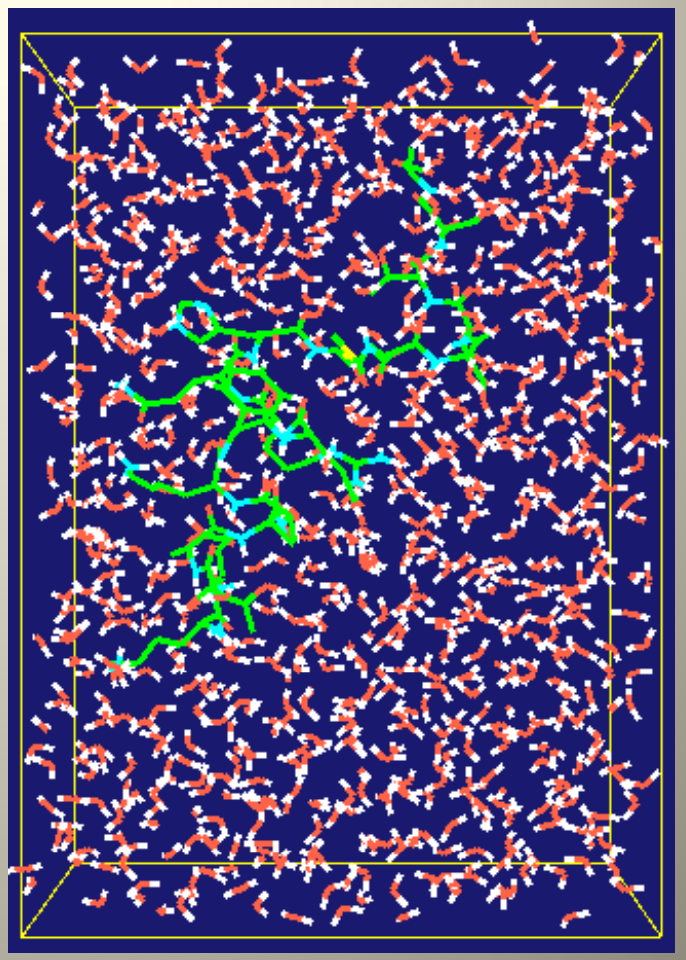


7. **full MD w.o. restraints**
   $ grompp -v -f full -o full -c after_pr -p speptide → **full.tpr** **mdout.mdp(modified)**
   $ mdrun -v -s full -e full -o full -c after_full -g flog >& full.job & → **full.edr** **after_full.gro** **full.job state_prev.cpt** **flog.log full.trr**

Ribonuclease S-peptide



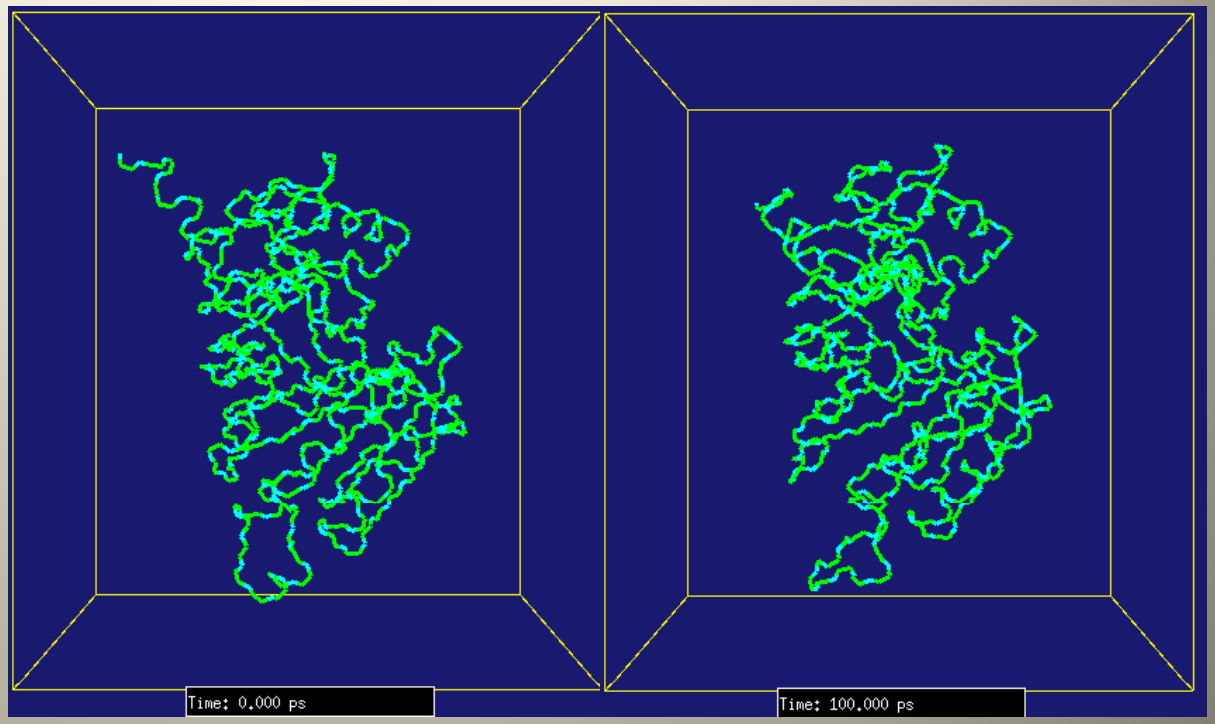
initial



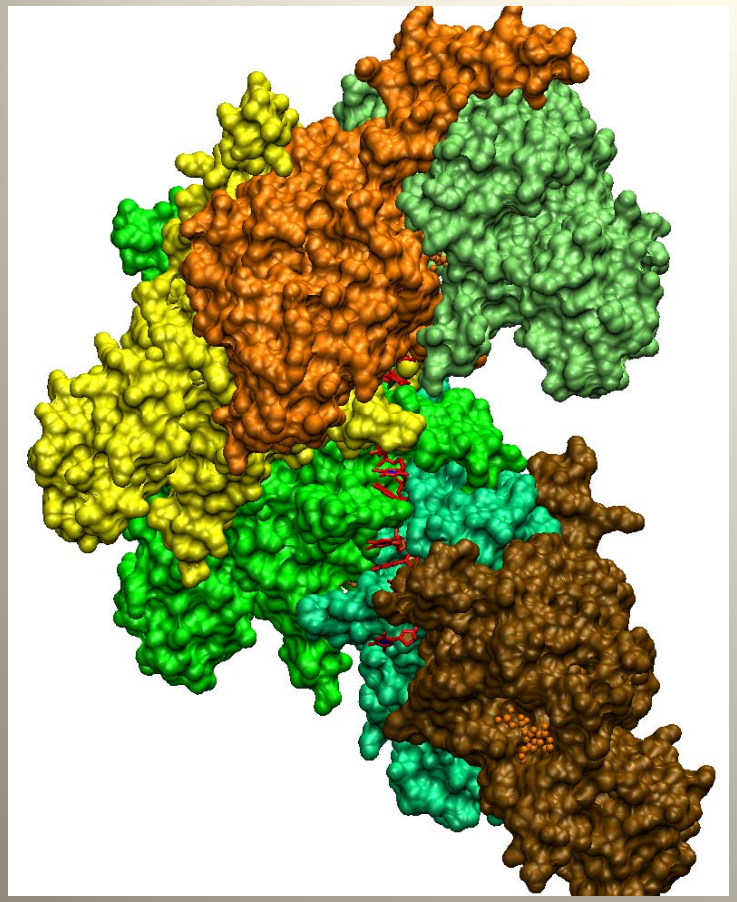
100ps

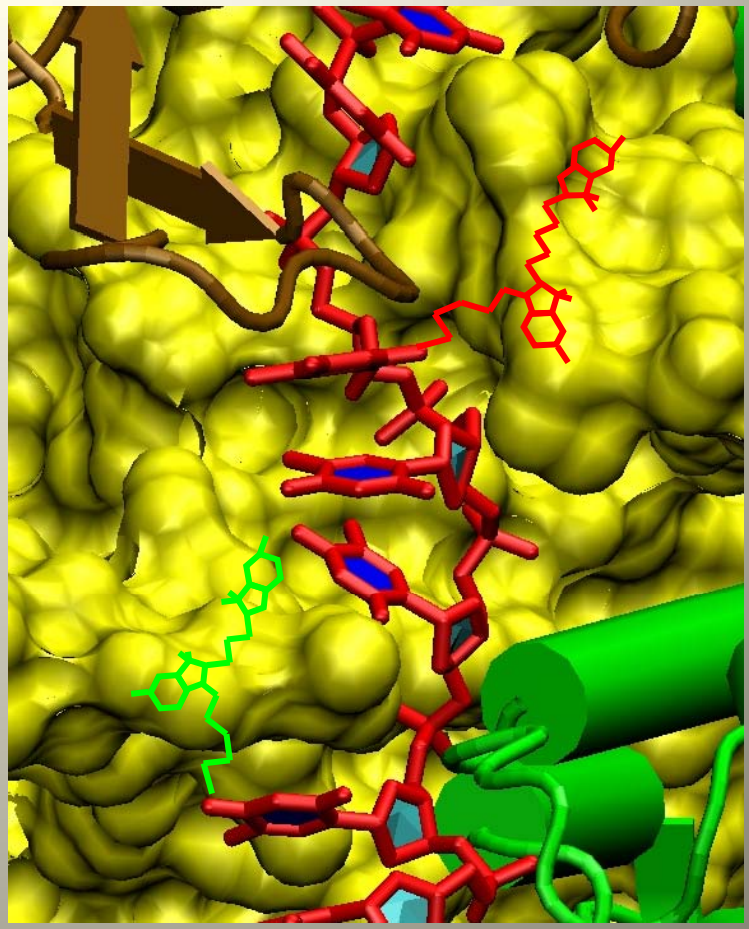
ngmx –s pr –f full

RecA

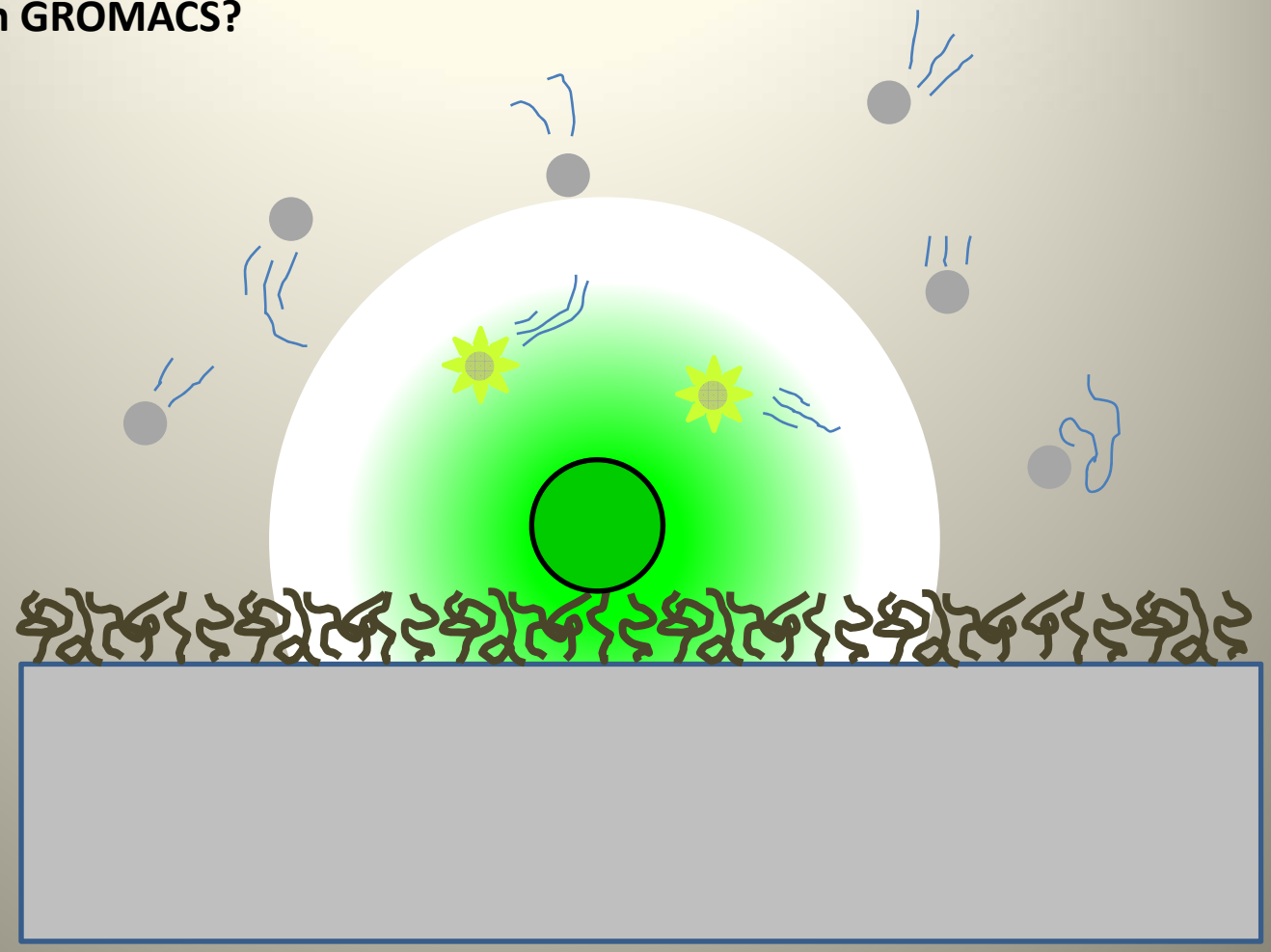


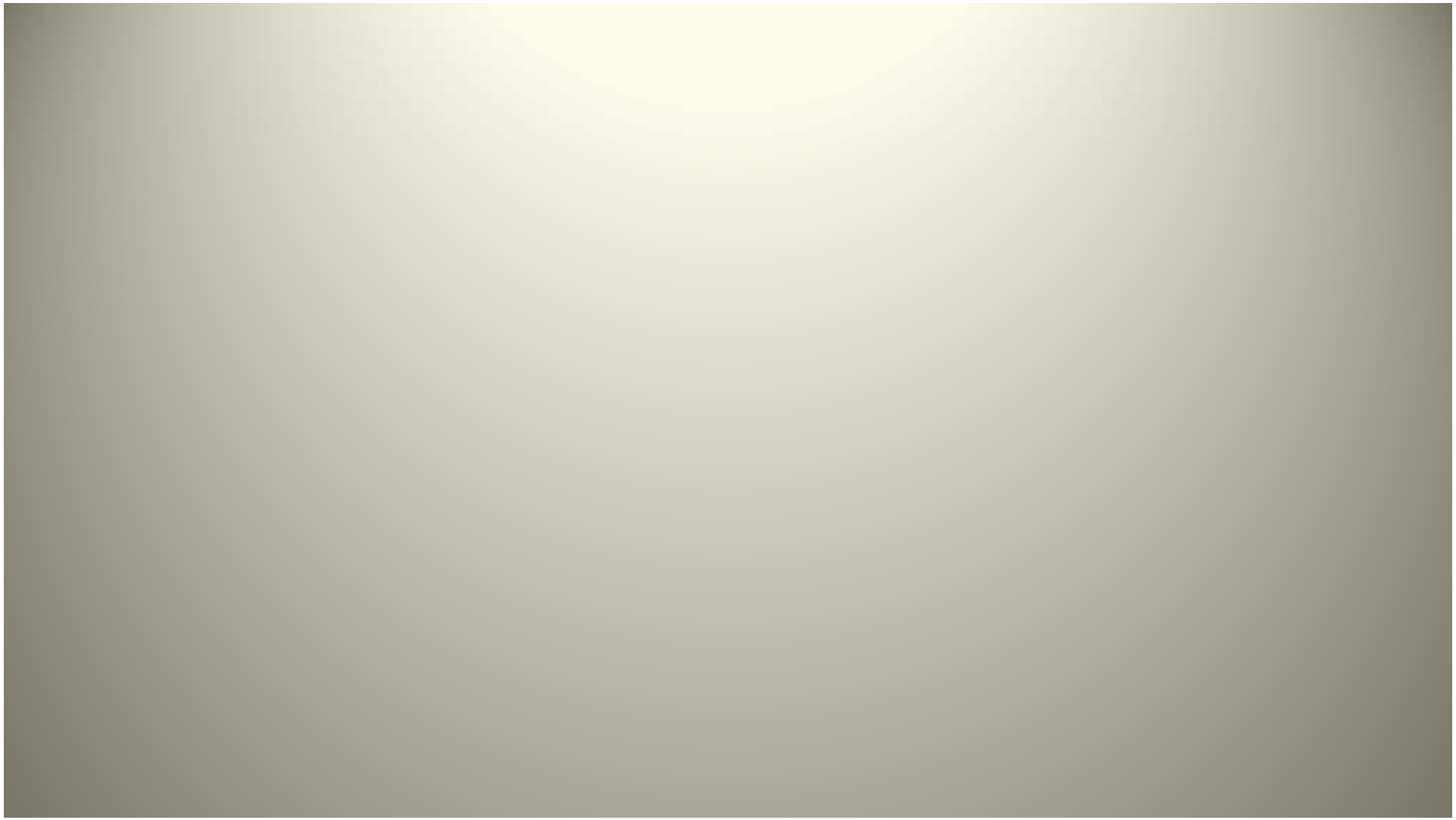
What do I want with GROMACS?





What do I want with GROMACS?





.gro file format

```
1WATER OW1 1 0.126 1.624 1.679 0.1227 -0.0580 0.0434
1WATER HW2 2 0.190 1.661 1.747 0.8085 0.3191 -0.7791
1WATER HW3 3 0.177 1.568 1.613 -0.9045 -2.6469 1.3180
2WATER OW1 4 1.275 0.053 0.622 0.2519 0.3140 -0.1734
2WATER HW2 5 1.337 0.002 0.680 -1.0641 -1.1349 0.0257
2WATER HW3 6 1.326 0.120 0.568 1.9427 -0.8216 -0.0244
```

Force calculation

```
For (i=1: I < #ptcl :i++){
For (j=i+1;  j < #ptcl : j++) {
   xr=x(i)-x(j);
   xr=xr-box*nint(xt/box)       /* PBC */
   r2=xr^2
   if(r2<rc2){
      r2i=1/r2;
      r6i=r2i^3;
      ff=48*r2i*r6i*(r6i-0.5);   /* derivative of LJ-potential */
      f(i)=f(i)+ff*xr;           /* Fx=-dV/dx=-dr/dx*dV/dr */
      f(j)=f(j)-ff*xr;
      en=en+4*r6i*(r6i-1)-ecut;
   }
}
}
```

Verlet Algorithm

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\dddot{r}}{3!}\Delta t^3 + O\left(\Delta t^4\right)$$

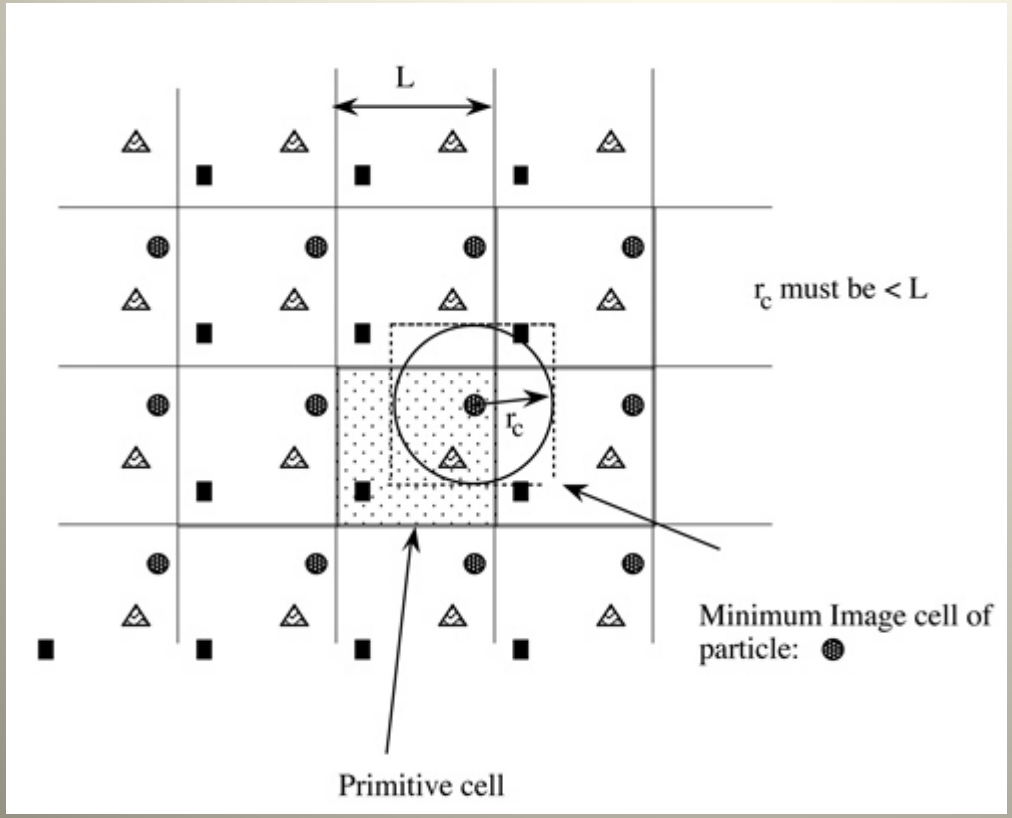
$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\dddot{r}}{3!}\Delta t^3 + O\left(\Delta t^4\right)$$

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \frac{f(t)}{2m}\Delta t^2 + \cancel{O\left(\Delta t^4\right)}$$

$$r(t+\Delta t) - r(t-\Delta t) = 2v(t)\Delta t + O\left(\Delta t^3\right)$$

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \cancel{O\left(\Delta t^2\right)}$$

Periodic boundary condition



http://epress.anu.edu.au/sm/html/ch06.html