

MiCMoS

Milano Chemistry Molecular Simulation

Description and User Manual

Tutorials

T12. MD simulation of organic nanoparticles

PURPOSE: In this tutorial, it is described how to prepare isolated (non-periodic) simulation boxes that include either free or solvated nanoparticles of small organic molecules.

REFERENCE to main manual: Sections 5.5 (*Excbox*), 5.6 (*Nanosolv*), 8.9 (*Nanocut*), plus Sections referenced in Tutorials T3, T7, T10 and T11.

FILES: Click here to download the materials for this exercise:

https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/meohmco.dat
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/pyridcut.dat
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/meohmc.pri
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/meoh.top
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/meoh.sla
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/pyrid.top
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/nano.top

The Tutorial consists of four different exercises. First, we will build a model for liquid methanol, following the instructions given in Tutorials T3 and T7 (you can skip this part if you feel confident with the Monte Carlo procedure). Second, we will simulate and analyze an isolated nanoparticle of pyridone. Third, we will compute the same nanoparticle embedded in a cubic nanodroplet of methanol. Finally, we will soak the same nanoparticle in a roughly spherical nanodroplet of methanol. The step-by-step procedure is illustrated from scratch.

Part 1. Equilibrating from scratch bulk liquid MeOH

The least straightforward task is to build a reasonable model (topology file + simulation box of equilibrated liquid) for the solvent. However, you can do it in a few minutes by following the instructions of Tutorials T3 and T7. It is a useful exercise to recap now the various steps in the correct order. The first task is to use **Retcor** as a molecule builder to generate a starting *.oih* file for methanol. Start exactly as explained in the Tutorial T3, opening an empty *meoh.oih* file and writing down the standard heading:

```
# Methanol isolated
0
1.0000    1.0000    1.0000    90.00    90.00    90.00
0.0
3
```

Remember that only the title is formatted (1x, 10a4); the format of the other lines is free. The second line is a dummy 0, the third expresses the crystallophysical reference system, the fourth the crystallization enthalpy (0.0 as we are not building a crystal structure) and the fifth

one the number of atoms that are explicitly defined. Remember that you always need three non-linear atoms to define the basic backbone of molecule 1; the other atoms will be defined implicitly according with Figure 1.2 and Table A2.1 in the Manual. A suitable (not unique) choice is the fragment O-C-H, where H is any of the three methyl hydrogens. The following instructions build these three atoms in the x,y plane so that the central carbon, C2, is tetrahedral (angle O-C-H \sim 110 deg).

```

1  0.00000  1.42500  0.00000  1 29  0.000
2  0.00000  0.00000  0.00000  1 13  0.000
3  1.01161 -0.37822  0.00000  1  3  0.000

```

Codes 29, 13 and 3 assign atoms identities according to Table 1.1. You can easily verify by yourself that the two bond distances are 1.425 Å (C-O) and 1.080 Å (C-H). Note that the three charges are 0.0, as they are unknown.

Next, you have to define the three remaining hydrogens as implicit atoms. You can complete the methyl by adding hydrogens 4 and 5 (atom type 3, Table 1.1) at once in "methylene mode" (Figure 1.2 and Table A2.1 in the Manual). The hydroxy H6 atom (atom type 5, Table 1.1) will be added exploiting the "Z-matrix" mode (see also Tutorial T3 for more details). This means that you have only 2 lines to add:

```

2
4      5      0      2      3      1      1      3      0.00    1.08      0.0      109.0
6     -1      0      1      2      3      1      5      0.00    1.00    180.0      106.0

```

Note that the torsion of 180 deg assigned to H6 staggers this atom with respect to H3, as it should be in a stable conformation. Moreover, the H-O bond is shorter (1.0 Å) than the C-H ones. To close the oih file you need to specify that the only symmetry operation is the identity (no translations) and that you don't need to redefine the origin. The last parameters are straightforward, and you can copy-and-paste them safely from the exercise of Tutorial T3. Finally, your meoh.oh file should be like:

```

# Methanol isolated
0
1.0000  1.0000  1.0000  90.00  90.00  90.00
0.0
3
1  0.00000  1.42500  0.00000  1 29  0.000
2  0.00000  0.00000  0.00000  1 13  0.000
3  1.01161 -0.37822  0.00000  1  3  0.000
2
4      5      0      2      3      1      1      3      0.00    1.08      0.0      109.0
6     -1      0      1      2      3      1      5      0.00    1.00    180.0      106.0
1
1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0
0.0000 0.0000 0.0000
0
0

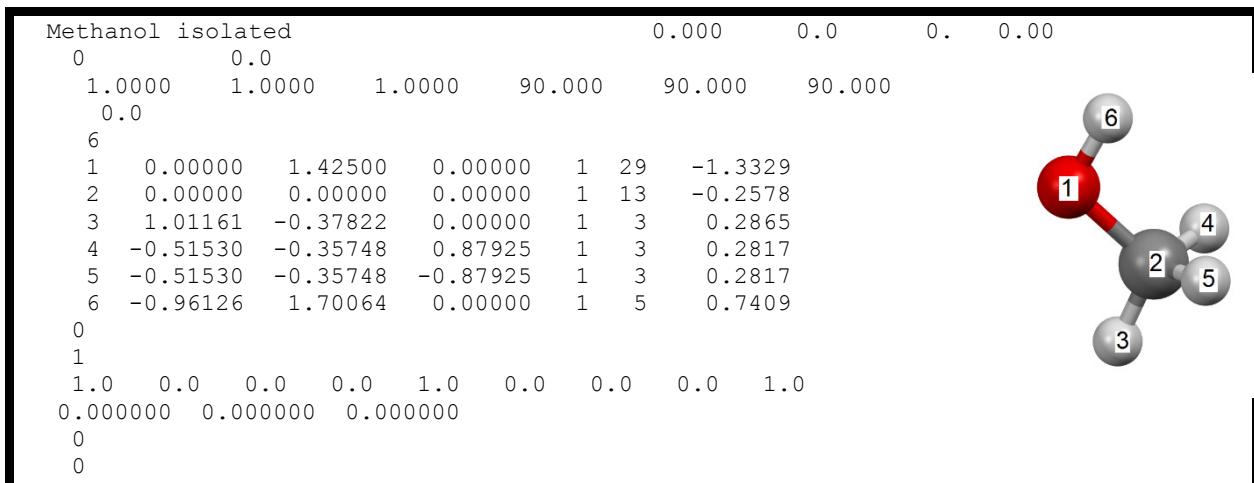
```

Now execute **Retcor** and **Retcha** to produce a .oeh file with the correct Hückel charges, suitable for the AA-CLP force field. Note that coordinates and charges of CLP methanol are also available in our MiCMoS library of topologies; you could use these data and build the meoh.oeh file by editing any existing .oeh file with the correct information. Check at https://sites.unimi.it/xtal_chem_group/index.php/topology-files/30-topologies. You might

choose also to follow the procedure detailed in Tutorial T2 to produce ESP-charges, if you prefer to apply the AA-LJC force field.

```
./run.retc当地 meoh
./run.retcha meoh
```

The procedure will end in the following `meoh.oeh` file (you can check the final results by checking the appearance of the molecule in Schakal: just open the `meoh.dat` file from **Retcor**, or edit any existing `.xyz` file by copying the Cartesian coordinates and open it with Mercury):



A model for the equilibrated liquid can be efficiently produced by Monte Carlo. To do so, you have to build a slave file to define the collective degrees of freedom that are going to be varied. In liquid methanol the business is very simple, as you have just 1 rotatable C–O bond. In other words, the molecule will be completely rigid apart the H6–O1–C2–H3 torsion. As usual, refer to Table 6.2 in the main manual to enter the correct quantities in `.sla` and `.top` files.

Start opening an empty `meoh.sla` file and typing the number of implicit groups (1). Then, add a new line and proceed by adding the geometrical parameters in Z-matrix mode. You need the H–O distance (1.00 Å), the H–O–C angle (110.0 deg) and the torsion with respect to any of the three methyl hydrogens: for example, if you choose H5, +60 deg allows you to start from a staggered conformation. If you choose H3, +180 deg produces the same conformer (check the correct parameter with a graphic program, like Mercury, Schakal or Diamond). Add the code 0 0 4 to specify that you want to change neither the bond distance, nor the angle, while the torsion will be varied by exploiting a maximum random step as specified by the 4th entry in the control vector (line #8 of the Monte Carlo control file `.mci`, see Section 6.2 and the discussion). Complete the line with 6 0 0 to generate only one atom, the 6th one:

```
1
1.00 110.0 60.0 0 0 4 6 0 0
```

Save the slave file and execute **Pretop** to generate a starting topology:

```
./run.pretop meoh
```

The program reads the `meoh.oeh` and `meoh.sla` files; answer 1 1 1 when prompted to give the scaling constants for the force constants.

Methanol: starting top file

```

Methanol isolated topology
 6
 1 0.00000 1.42500 0.00000 29 -1.3329
 2 0.00000 0.00000 0.00000 13 -0.2578
 3 1.01161 -0.37822 0.00000 3 0.2865
 4 -0.51530 -0.35748 0.87925 3 0.2817
 5 -0.51530 -0.35748 -0.87925 3 0.2817
 6 -0.96126 1.70064 0.00000 5 0.7409
 1 nslav-u
 6 0 0 0 0 0 0 0.0000
 0 ncore-v
 0 nslav-v
 36.0 0.0 volu-u,volu-v
 5 nstr-u
 1 2 1.425 3530.0 O- C
 1 6 1.000 4250.0 O- H
 2 3 1.080 3600.0 C- H
 2 4 1.080 3600.0 C- H
 2 5 1.080 3600.0 C- H
 0 nstr-v
 7 nbend-u
 1 2 3 110.00 630.0 O- C- H
 1 2 4 109.00 642.5 O- C- H
 1 2 5 109.00 642.5 O- C- H
 2 1 6 106.00 450.0 C- O- H
 3 2 4 109.00 470.0 H- C- H
 3 2 5 109.00 470.0 H- C- H
 4 2 5 109.00 470.0 H- C- H
 0 nbend-v
 1 ntors-u
 6 1 2 3 50.00 -1.0 3.0 H- O- C- H
 0 ntors-v
 0 nlist-u
 0 nlist-v
 0.410 235.0 650.0 77000.0
 0 nextr

```

Methanol: top file after editing

```

Methanol isolated topology
 5
 1 0.00000 1.42500 0.00000 29 -1.3329
 2 0.00000 0.00000 0.00000 13 -0.2578
 3 1.01161 -0.37822 0.00000 3 0.2865
 4 -0.51530 -0.35748 0.87925 3 0.2817
 5 -0.51530 -0.35748 -0.87925 3 0.2817
 1 nslav-u
 6 -1 0 1 2 5 5 0.7409
 0 ncore-v
 0 nslav-v
 36.0 0.0 volu-u,volu-v
 0 nstr-u
 0 nstr-v
 0 nbend-u
 0 nbend-v
 1 ntors-u
 6 1 2 5 2.0 1.0 3.0 H- O- C- H
 0 ntors-v
 0 nlist-u
 0 nlist-v
 0.410 235.0 650.0 77000.0
 0 nextr

```

A tentative `meohtry.top` file is printed; copy it into `meoh.top` and open this last file. As usual (see Tutorials T7 and T8) you will need to delete the lines highlighted in red and edit the green ones. First, complete the slave line: you must specify here the connectivity, identity and charge of the implicit group. As you are defining the 6th atom in Z-matrix mode (Table 6.2), `n1=6` (already given in the tentative `top` file), `n2= -1`, `n3=0`, while `n4`, `n5` and `n6` specify the torsion-defining atoms at which H6 is attached: `n4=1`, `n5=2`, `n6=5`. Remember that this line replaces the explicit definition of atom 6 as the last entry in the coordinates stream; use the atom identity (5) and charge (+0.7409) to complete the slave line. Then, delete the corresponding line and update the atom count (5 explicit atoms, not 6). Set 0 stretching potentials (0 str-u) and delete the next 5 lines; set 0 bending (0 nbend-u) and delete the next 7 lines. Finally, you have to define the correct parameters for the H6-O1-C2-H5 torsion. To favor staggered conformers, you must choose a potential that has 3 equivalent minima at +60 deg, +120 deg and +180 deg. If you look at Figure A7.1 in the Manual Appendix, you will see that the form that exactly fits your need is the one with $f = +1$ and $m = 3$, so you have to change -1 into +1 in the torsion line. Moreover, a force constant of +50 kJ/mol is strongly overestimated. The Me–OH bond is highly rotatable; the correct value for methanol is given in Table A7.5 (molecule 42) and reads 2.0 kJ/mol.

The next task is to produce the liquid box. Execute **`Boxliq`**.

```
./run.boxliq meoh
```

The program reads the `meoh.oeh` and `meoh.sla` files. Answer 7 1.5 0.5 10.0 0 when prompted; the first entry (7) tells the program to build 7x7x7 replicas of the starting cell; 1.5 is the expansion factor to reduce the occurring of hard contacts; 0.5 and 10.0 are parameters for the random generation of conformers, and the last 0 specifies that all the molecules in the liquid have the same chirality. With these parameters, you have a cubic box containing 1024 molecules with a density of 0.622 g/cm³ and an edge of 44.4 Å:

```
min and max x,y,z translations -22.339 22.081 -22.333 22.088 -22.336 22.083
box on x, y, z 44.4199 44.4203 44.4191 averaged to 44.4198
normal end of operation, n.mols., dens 1024 0.622
Thank you for using MiCMos
```

Eventually, let's write an input command file, `meoh.mci`, to tell the MC engine what to do. The best way is to take the file you already employed in Tutorial T7 (liquid pentane) and to edit it according to our current needs. Copy `pentane.mci` into `meoh.mci` and edit the file as shown below; the necessary changes are highlighted in green.

```
MeOH liquid run
# iprint ivarib iwrh ipots factin
    0      0      0      0      0.7
# cutoff boxx  boxy  boxz  alf  bet  gam  var.indices      irbox
  16.0  0.000  0.000  0.000  0.0  0.0  0.0  2  2  2  0  0  0  1
# temp n.moves,ncom reset,nbox reset nwri nwre npri/steps
  298.0  1000000  1000      0      0      0      2000
  0.70  50.0  10.0  5.0  0.0  0.  0.  0.  0.  0.
# P npres ianis
  1.0    700      3
```

Apart modifying the title, we have raised T to 298 K (otherwise, the algorithm would operate at 20 K, acting as an energy minimizer) and we have increased the number of steps to 10^6 to

ensure a reasonable equilibration. We added also the fourth entry (5.0) in line #8, where the maximum step widths for random changes are given. This is necessary, as in the `meoh.sla` file we explicitly required to use the 4th slot to define the maximum step width for the H-O-C-H torsion. The other parameters have all the usual meaning (see Tutorials T7 and T8 and Section 6.6.2 in the Manual): we do not want to generate bulky files, thus we do not require that the trajectory (extension `mcc.dat`) and energy (`mcc.ene`) files are printed (`nwri=nwre=0`).

CAUTION: if you want the full output, ensure that `nwre` and `nwri` parameters be `>0`.

In 20-40 min depending on your clock speed, the calculation will end; if you want to skip the dead time, you can download the final frame of the trajectory, `meohmc.dat`, and the corresponding MC output, `meohmc.pri`, from our repository:

https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/meohmc.dat

https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/meohmc.pri

If you have a look at the MC output, you can see that, after disposal of hard contacts that make the initial box swelling, the system converges to a final density of 0.753 g·cm⁻³, reasonably close to that of liquid methanol (0.792 g·cm⁻³). This is a reasonable starting model for the liquid, though still not completely equilibrated yet. The final cubic box has an edge of 41.6678 Å: this will be the maximum dimension of the nanodroplet we are going to build.

Part 2. Isolated nanoparticle of pyridone

While you are waiting for the the simulation of liquid MeOH to finish, you can carry on preparing the nanoparticle. We will keep working on pyridone. Start from the same `.cif` file of Tutorial T10, `PYRIDO11.cif`, and dorectly prepare the `.oeh` file with the macro `run.cry`:

```
./run.cry PYRIDO11
```

When prompted, normalize hydrogens, ask for a normal output and perform H connectivity checks. The macro calls in sequence ***Retcif***, ***Retcor*** and ***Retcha*** and goes straight to a full `PYRIDO11.oeh` file, getting rid of unnecessary files. Rename `PYRIDO11.oeh` into `pyrid.oeh`, which will be the input for ***Nanocut***. Before, you must define input parameters to generate the nanoparticle. Open an empty `pyrid.inp` file and type in the following instructions:

```
# ipack1, ipack2, ipack3 (initial cluster)
 2      2      2
# iprint (=0 normal, 1=extended printout)
 0
# nplanes (number of (hkl) planes for boundary conditions)
 4
# h,  k,  l,  dhkl
 2  0  0  8.0
 1  0 -1  8.0
 1  0  1  8.0
 1  1  1  8.0
```

Refer to the main Manual, Section 8.9, for full explanations. Remember that the specified ipack factors correspond to a 5x5x5 starting box, as elementary translations run from \pm ipack each ($2 \cdot$ ipack+1 translations for each edge). This box is then cut accordong to the boundaries defined by the 4 (nplanes) crystallographic planes specified below (plus their centrosymmetric counterparts).

The last input parameters dhkl express the distance of each plane from the origin of the coordinate system, which lies in the central cell. We want to ensure that the cluster is fully solvated, thus we keep the the dimensions of the prism well below the edge of the simulation box of MeOH (41.7 Å, see above). Note that playing with dhkl you can shape the nanoparticle according to your needs.

CAUTION. This is a purely illustrative example. We do not know whether the selected facets correspond to a minimum of the total surface tension according to the Wulff's theorem. What is worse, it is impossible to know whether the morphology of a pyridone nanocluster be identical to that of a macroscopic crystal.

Save the pyrid.inp file and execute **Nanocut**.

```
./run.nanocut pyrid pyrid
```

The first entry corresponds to the .inp file, the second one to the .oeh file. In a fraction of second the program prints the following files: pyridcut.pri contains a report on how the cluster has been generated; pyridcut.dat and pyridcut.xyz are the Cartesian coordinates of the generated cluster, either in MiCMoS .dat format or in .xyz format. The latter can be visualized by Mercury as is; the former can be converted into .gro format by **Datgro** (see also Tutorial T11). Have a look at pyridcut.pri. The program summarizes the input parameters, checks the atom identities and symmetry operations and builds the cluster. For each molecule, its centre of mass vector is projected onto unitary reciprocal directions $\mathbf{s}_{hkl}/|\mathbf{s}_{hkl}|$, where hkl are the reciprocal lattice directions of the planes defined in the .inp file: if the projection is greater than dhkl, the molecule is discarded. The real and reciprocal components of the **s** vectors are shown before the final molecule count. As for the present exercise, 463 molecules in the starting box are discarded and the final cluster consists of 37 molecules only.

Now, copy the topology file of pyridone (pyrid.top) that you employed in Tutorial T10, or dowload it from:

https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T12/pyrid.top.

The next step is to prepare the MD command file for the nanoparticle. Open an empty file pyrid.mdi and type in the following instructions. You can also copy the .mdi file you already used in Tutorial 10 and update it according with our current needs. Critical instructions are highlighted in green:

```
pyridone nanoparticle clp
# n.steps irvel ipri ibox idstr timestep Emolim iengt ibias +Ebias Nbias tinon tinof
 10000      0      0      0      0      0.002      -5.0      0      0
# cutoffu cutofffv cutofffuv factin ipots ianh inano
 15.0      0.0      0.0      0.0      0      0      0
# N(T) Tset Tstart Trelax 0/1/2 weak/stiff/CSV
 250      298      298      0.6      0
# N(P) Pset comprs 0/1ianis ipr ww iextstr + strall 22, 33, 12 13 23, GPa
 0      0.0      0.0      0      0.0      0
# N(com) nwbox nwre npri
 50      200      200      100
# Nrot-ev romin maxs stepr Rev fact icub
 50      -15      11      3      25.0      0.9      2
```

First, the calculation is non-periodic now. This means that ibox must be equal to 0 in the first input line. For this reason, no pressure control is required: N(P)=0. Moreover, ibox=0 requires

that you add the `#Nrot-ev` line, which specifies the parameters for suppression of rotational momentum of the cluster and the strategy for dealing with evaporated molecules. See Sections 7.5 and 7.6.2 of the manual for a full description of the procedure; we note here that `icut=2` implies that the tethering algorithm is active, *i.e.*, any molecule far more than 25 Å from the overall centre of mass is “tethered” back by rescaling this distance by a factor of 0.9. At the same time, the best back-rotations of the whole cluster are found by exploring 11x11x11 possible rotation matrices ranging from -15 deg to +15 deg in steps of 3 deg around the X, Y, Z axes. Every 50 moves, the matrix that minimizes the distances between current centres of mass and the starting configuration is chosen to restore the original orientation.

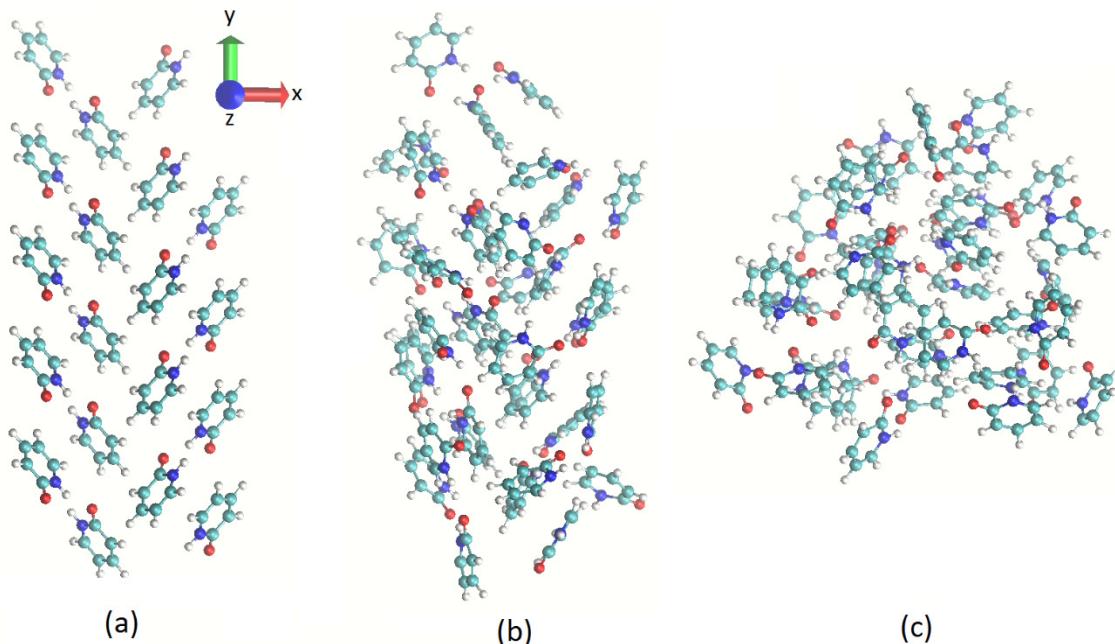
When ready, you can launch the MD simulation:

```
./run.mddomain pyrid pyridcut pyrid
```

Where, as usual, the first name (pyrid) is the name of both topology and instruction (`.mdi`) files, the second one (pyridcut) that of the simulation box and the third (pyrid) specifies the identities of the output files.

The simulation will end in a few minutes. If you convert the trajectory in `.gro` format with **Datgro** (see Tutorial T11 and Section 8.6 in the main Manual), you can visualize the time evolution of your system with VMD®. The cluster loses very quickly its starting and symmetry. In a few picoseconds, it melts, achieving an amorphous state.

Molecular Dynamics evolution of an isolated nanocluster of pyridone (37 molecules) at T = 298 K, as seen down the Z laboratory axis. (a) t = 0.0 ps. (b) t = 0.8 ps. (c) t = 20.0 ps.



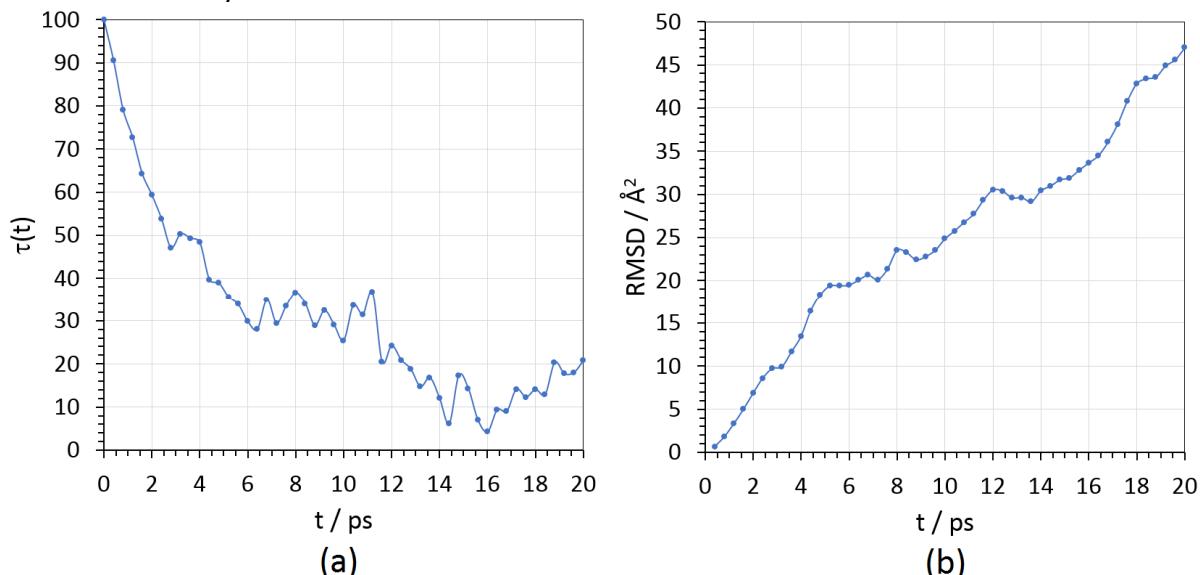
This phenomenon can be appreciated also by inspecting at the rotational correlation function, which can be computed by the module **Correl** (see Tutorial T11 and Section 8.4 in the main manual):

```
./run.correl pyridmdc
```

Answer "1 51 0.002" for the reference frame, total number of frames and timestep. Then, define a vector perpendicular to the main ring of the molecule using the "1 5 4 7 1" string when prompted (see the Manual for explanations). Finally, answer "0 20.0" to explore the molecular centre of mass with a cutoff distance of 20 Å. You can plot the results with any graphical interface and compare them with the Literature (for example, see A. Gavezzotti and L. Lo Presti, *New J. Chem.*, 2019, 43, 2077-2084).

As expected, the rotational correlation is reduced to 10-20 % in the first 20 ps of the simulation. However, it does not go exactly to zero as quickly as in a true liquid (see for example the results on liquid benzene in Tutorial T11): a residual order must persist, even though it is probably appreciable only at short range. Accordingly, the average RMSD of the molecular centers of mass steadily increases, but slowly if compared with a liquid.

(a) *Rotational correlation coefficient for the pyridone nanoparticle (125 molecules) as a function of t.* (b) *Average root mean square displacement for the molecular centers of mass in the same nanoparticle.*



You can analyze the final frame of the trajectory with ***Analys*** (see Tutorial T11 and Section 8.2 in the main Manual). You can give the whole trajectory (*pyrimdc.dat*) as input and fill in the various fields as follows:

```
./run.analys pyrid pyridmdc
+++++++
MiCMoS 2.2 jul 2023 ++++++
This is the New-Analys module

Give start and final frame number,=1,1 if one frame
51 51

check short intermolecular distances
give % of contact R, e.g.90., zero to skip
0
g(R) calculations, 0=yes, 1=no
0
give pack coeff if box not present,0.5liq-0.7cry
0.7
c.o.m. radial distribution function
give rmin,step,n.bins, zeros to skip
2.0 0.1 50
```

```

atom-atom RDFs, INTERMOLECULAR ONLY
give rmin,step,n.bins max 300,zeros to skip
1.0 0.05 120
give ncpu,number of solute rdf, 0 skip, -1=do all
1
now give ncpu pairs of atom types ***i-j with i<j***
7 27
give ncpv,number of solvent rdf, 0 skip -1 do all
0
give ncpuv,n. of solute-solvent rdf,0 skip -1 all
0
radial function smoothing, 0=yes, 1=no
0
Skipping frames...
dat file is trajectory at MD st solutes and solvents      37      12      0      0
frame n.      51 moves      10000
mol.vol.u, vol.v, approx cpack, Vbox  88.1    0.0    0.70    4656.7
normal end of operation at frame n.      51
Thank you for using MiCMos

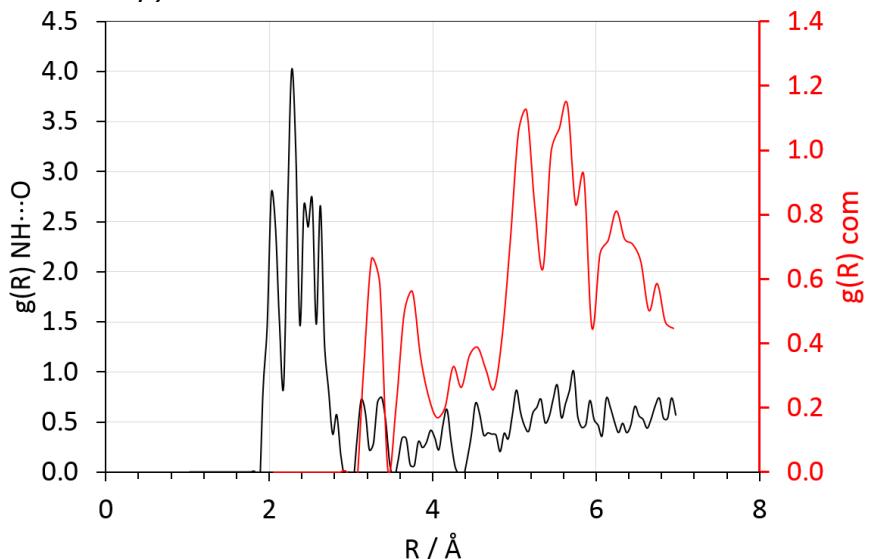
```

Note that you could have obtained the same results asking to analyze the last frame (*pyridmdo.dat*) and indicating "1 1" as the initial and final frame.

CAUTION. It is important here to specify a value for the packing coefficient (cpack) when asked. We gave 0.7, which is reasonable for crystals. If cpack=0.0, the program fails to correctly normalize the $g(R)$ function for molecular centers of mass.

The graphical result is shown in the next Figure. The $g(R)$ for centres of mass is shown in red, that for the $\text{NH}\cdots\text{O}$ intermolecular hydrogen bond in black. Compare the plot with those computed for the pyridone crystal (Tutorial 11): the peaks are much less structured now, due to the melting of the nanoparticle. The c.o.m. RDF loses its two intense peaks at ~ 4 and ~ 6 \AA . The $\text{NH}\cdots\text{O}$ hydrogen bond survives, but the modulations at high R are completely lost, indicated that the long-range structure is destroyed. Interestingly, a residual peak around 2-2.5 \AA is still present, though with lower intensity, indicating that some pyridone dimers are likely present in the liquid-like amorphous state.

Radial distribution fuction, $g(R)$, for the last frame ($t = 20 \text{ ps}$) of the dynamical time evolution of a nanocluster of 37 pyridone molecules.



Part 3. Solvated pyridone nanoparticle

In the third part of this Tutorial, we will check what happens if the same nanoparticle is solvated. The MC simulation on the bulk methanol should be complete now: check that you obtained the correct liquid density (see above) and launch **Nanosolv** (Section 5.5 in the main Manual):

```
./run.nanosolv pyridcut meohmco
```

The first name corresponds to the starting *.dat* file of the nanoparticle; the second one is the last frame (in Cartesian coordinates) of the MC trajectory for 1024 methanol molecules. The program prompts you to give a tolerance factor (atol). As explained on screen, this serves to tune the solute-solvent contact distance that determines whether a solvent molecule is erased or not. To avoid hard contacts, the check is done on individual atom-atom pairs among solute and solvent molecules. Usually, an atol parameter of 0.0 is enough, but you can increase it or even input a negative number, if you want to start keeping the solvent and the nanoparticle closer.

```
*****
MiCMoS v2.2
July 2023
This is the Nanosolv module
*****
Give the contact tolerance, atol, in Angstrom
The minimum allowed contact distance will be RvdW(i)+RvdW(j)+atol
```

Answer 0.0 for the moment. The program will echo your input and will print some information on screen. In a few seconds you will obtain the results:

```
Merging frames
It may take a while, depending on the number of mols
...please be patient...

Done

Found      160  molecules in the liquid box within clash limits

Finalizing output...

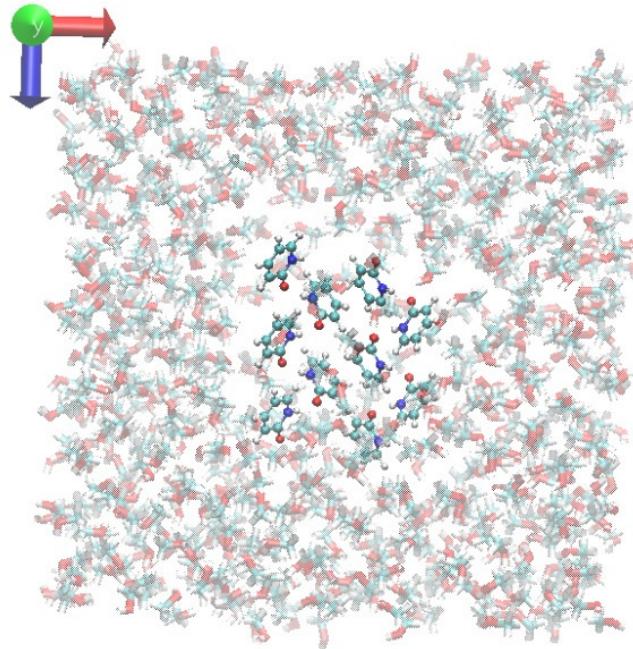
Done

Written      864  solvent molecules in merged .dat file

Normal program termination
Thanks for using MiCMoS
```

Essentially, **Nanosolv** puts the nanoparticle at the centre of the liquid box, replacing solvent molecules at van der Waals distance with any of the solute atoms. As for the present exercise, it finds 160 molecules in clash and removes them. The other 864 MeOH molecules form your solvation shell. We used again **Datgro** and VMD to check the starting model, which can be considered satisfactory.

*Nanoparticle of pyridone (37 molecules) embedded in 864 molecules of methanol: starting point from **Nanocut**.*



To start the simulation, you need to build the topology file for pyridone *and* methanol. To do so, you have to merge the pyrid.*top* and meoh.*top* files into a unique topology file. Start by copying the solute topology into a new file, which will call nano.*top*, and open it:

```
cp pyrid.top nano.top
vi nano.top
```

Now, you must add pertinent information of the meoh.*top* file into nano.*top*. Obviously, methanol will become the “solvent” specie, while pyridone will be the “solute”. Refer to Section 7.6.4 in the main Manual. First, you must enter the solvent coordinates (ncore-v block of instructions). All the necessary information is saved either in the **Pretop**-generated meohtry.*top* file or in the MC topology, meoh.*top*, so you can easily copy-and-paste the needed parameters. In nano.*top* change

```
0 ncore-v
```

into:

```
6 ncore-v
1 0.00000 1.42500 0.00000 29 -1.3329
2 0.00000 0.00000 0.00000 13 -0.2578
3 1.01161 -0.37822 0.00000 3 0.2865
4 -0.51530 -0.35748 0.87925 3 0.2817
5 -0.51530 -0.35748 -0.87925 3 0.2817
6 -0.96126 1.70064 0.00000 5 0.7409
```

CAUTION. You can find the Cartesian coordinates and charges of MeOH in the correct format in the original meohtry.*top* file produced by **Pretop**, and not in the meoh.*top* one, as you made changes to insert slave atoms (see above). For the same reason, the try.*top* file will provide you with information on stretching and bending parameters. If you erased the try file by mistake, don't worry! You can regenerate it by running again **Pretop** on meoh.*oeh*.

Proceed with the stretching parameters. In *nano.top*:

```
0  nstr-v
```

Must become

```
5  nstr-v
1    2    1.425  3530.0  O- C
1    6    1.000  4250.0  O- H
2    3    1.080  3600.0  C- H
2    4    1.080  3600.0  C- H
2    5    1.080  3600.0  C- H
```

Similarly, for bending the instruction:

```
0  bend-v
```

Will become:

```
7  nbend-v
1    2    3    110.00   630.0  O- C- H
1    2    4    109.00   642.5  O- C- H
1    2    5    109.00   642.5  O- C- H
2    1    6    106.00   450.0  C- O- H
3    2    4    109.00   470.0  H- C- H
3    2    5    109.00   470.0  H- C- H
4    2    5    109.00   470.0  H- C- H
```

Finally, insert the torsions for the solvent. Remember that you had to update the unique torsion of MeOH (see above), thus is safer to take the corresponding parameters from *meoh.top*. The instruction:

```
0  tors-v
```

Must be changed into

```
1  ntors-v
6    1    2    5    2.00    1.0    3.0  H- O- C- H
```

The merged *nano.top* topology file looks like:

```
#PYRIDO nanoparticle in MeOH
12
1   0.01285  -1.12334  -0.25939  21   -0.4151
2   -0.01770  0.01349  -2.24689  27   -1.2925
3   0.00355  0.04787  -0.98767  10   0.9400
4   0.01763  1.24854  -0.21064  12   -0.3916
5   -0.00280  1.19329  1.15056  12   -0.1739
6   -0.01512  -0.03594  1.84148  12   -0.3637
7   0.00272  -1.17359  1.10323  12   0.0499
8   0.04471  2.20723  -0.70716  2    0.2994
9   -0.00977  2.11419  1.71467  2    0.2847
10  -0.03795  -0.07028  2.92073  2    0.2947
11  0.00897  -2.13194  1.60120  2    0.2845
12  0.02530  -1.97775  -0.77886  7    0.4835
0  nslav-u
6  ncore-v
```

```

1 0.00000 1.42500 0.00000 29 -1.3329
2 0.00000 0.00000 0.00000 13 -0.2578
3 1.01161 -0.37822 0.00000 3 0.2865
4 -0.51530 -0.35748 0.87925 3 0.2817
5 -0.51530 -0.35748 -0.87925 3 0.2817
6 -0.96126 1.70064 0.00000 5 0.7409
0 nslav-v
88.1 0.0 volu-u,volu-v
12 nstr-u
1 3 1.379 4924.3 N- C
1 7 1.364 5293.6 N- C
1 12 1.000 5300.0 N- H
2 3 1.260 7226.6 O- C
3 4 1.430 4033.3 C- C
4 5 1.362 5460.5 C- C
4 8 1.080 3600.0 C- H
5 6 1.410 4456.5 C- C
5 9 1.080 3600.0 C- H
6 7 1.356 5590.2 C- C
6 10 1.080 3600.0 C- H
7 11 1.080 3600.0 C- H
5 nstr-v
1 2 1.425 3530.0 O- C
1 6 1.000 4250.0 O- H
2 3 1.080 3600.0 C- H
2 4 1.080 3600.0 C- H
2 5 1.080 3600.0 C- H
18 nbend-u
1 3 2 120.00 702.0 N- C- O
1 3 4 115.00 562.3 N- C- C
1 7 6 121.00 618.1 N- C- C
1 7 11 120.00 505.0 N- C- H
2 3 4 124.00 745.7 O- C- C
3 1 7 124.00 646.0 C- N- C
3 1 12 117.00 460.0 C- N- H
3 4 5 121.00 590.8 C- C- C
3 4 8 120.00 505.0 C- C- H
4 5 6 122.00 597.8 C- C- C
4 5 9 119.00 517.5 C- C- H
5 4 8 120.00 505.0 C- C- H
5 6 7 118.00 569.7 C- C- C
5 6 10 121.00 492.5 C- C- H
6 5 9 119.00 517.5 C- C- H
6 7 11 120.00 505.0 C- C- H
7 1 12 119.00 460.0 C- N- H
7 6 10 121.00 492.5 C- C- H
7 nbend-v
1 2 3 110.00 630.0 O- C- H
1 2 4 109.00 642.5 O- C- H
1 2 5 109.00 642.5 O- C- H
2 1 6 106.00 450.0 C- O- H
3 2 4 109.00 470.0 H- C- H
3 2 5 109.00 470.0 H- C- H
4 2 5 109.00 470.0 H- C- H
12 ntors-u
7 1 3 4 50.00 -1.0 1.0 C- N- C- O
3 1 7 6 50.00 -1.0 1.0 C- N- C- C
1 3 7 12 100.00 -1.0 1.0 N- C- C- H
1 3 4 5 50.00 -1.0 1.0 N- C- C- C
3 1 2 4 100.00 -1.0 1.0 C- N- O- C
3 4 5 6 50.00 -1.0 1.0 C- C- C- C
4 3 5 8 100.00 -1.0 1.0 C- C- C- H
4 5 6 7 50.00 -1.0 1.0 C- C- C- C
5 4 6 9 100.00 -1.0 1.0 C- C- C- H
5 6 7 1 50.00 -1.0 1.0 C- C- C- N
6 5 7 10 100.00 -1.0 1.0 C- C- C- H
7 1 6 11 100.00 -1.0 1.0 C- N- C- H
1 ntors-v
6 1 2 5 2.00 1.0 3.0 H- O- C- H
0 nlist-u
0 nlist-v
0.410 235.0 650.0 77000.0
0 nextr

```

Save this file. Now, copy the same parameter file you employed for the simulation of the isolated cluster, `pyrid.mdi`, into `nano.mdi`, as the instruction file must have the same name of the topology file. Then, edit `nano.mdi`, as the presence of solvent requires some adjustments.

```
Pyridone-MeOH solv nanop clp
# n.steps irvel ipri ibox idstr timestep Emolim iengt ibias +Ebias Nbias tinon tinof
  2000      0      0      0      0      0.001     -5.0      0      0
# cutofffu cutofffv cutofffuv factin ipots ianh inano
  40.0      40.0      40.0      0.0      0      0      0
# N(T)    Tset Tstart Trelax 0/1/2 weak/stiff/CSVR
  50      298      298      0.6      0
# N(P)    Pset    comprs 0/1ianis ipr  ww    iextstr + strall 22, 33, 12 13 23, GPa
  0      0.0      0.0      0      0      0.0      0
# N(com)  nwbox    nwre    npri
  25      50      50      50
# Nrot-ev romin    maxs    stepr    Rev    fact    icut
  20     -15      11      3      35.0      0.9      2
```

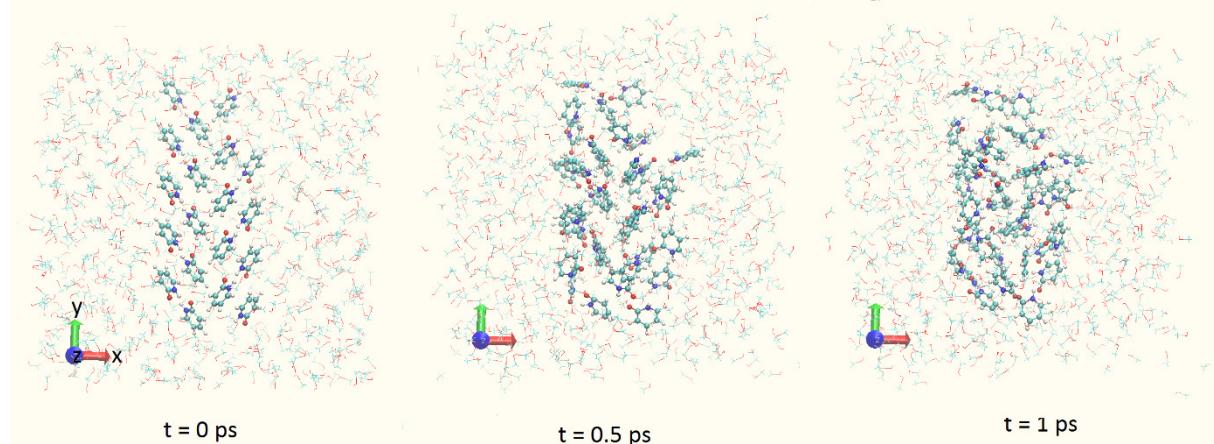
As usual, relevant instructions are highlighted in green. As the calculation is very long (~3-4 hrs for 10 ksteps), we simulate just the first 2 picoseconds of this system for demonstration purposes. Thus, reduce the number of steps to 2000. This will require ~ 45-60 min on a fast laptop. Reducing the time step to 1 fs can help to avoid divergence, especially when the system is far from equilibrium. Most important, it is mandatory to add cutoffs also for the solvent-solvent and the solute-solvent interactions: don't forget them, or the trajectory will become quickly very odd! As the simulation is not periodic, it is also wise to increase these cutoffs to ensure that all the relevant interactions contribute to the dynamics. The drawback is that the computational cost will be higher, as you might expect. Finally, it might be useful to increase the control frequency of thermostat and momentum suppression algorithms. This usually helps to maintain the kinetic energy in the correct range, while the high-frequency degrees of freedom remain correctly populated. In this run, we also increased the writing frequency of output files, to have smoother trajectories.

Execute the MD engine based on these `nano.mdi` instructions, using the `nano.top` topology and the `pyridcutsolv.dat` starting box:

```
./run.mddmain nano pyridcutsolv nanosolv
```

To avoid overwriting the results for the isolated cluster, the prefix of the output files will be "nanosolv" this time. The next picture shows some snapshots of the trajectory, taken from VMD after having converted the `mdc.dat` file into `.gro` format with **Datgro**:

Snapshots of the MD trajectory computed on pyridone nanocluster (37 molecules) in MeOH.



The time scale of this simulation is very different than that on the isolated cluster: it is difficult to reach conclusions on the stability of the cluster by analyzing just the first picosecond of the trajectory. We can say that the starting crystal structure of the cluster is lost very quickly also in this case. Interestingly, the crystal structure collapses before the solvent can destroy the pyridone nano-aggregate (pyridone is soluble in methanol). This results could also depend on the dimensions and shape of the nanoparticle and even on the starting state of the liquid; a systematic study like that published in A. Gavezzotti and L. Lo Presti, New J. Chem., 2019, 43, 2077-208, would shed light on this matter. To further analyze this system, you could compare the output of **Analys** taken at the same time frame for the two trajectories: you should be able to do it by yourself now.

Part 4. Building a roughly spherical nanodroplet of MeOH

Another point that requires attention is that, for the moment, we simulated a *cubic droplet* of methanol. This means that the system must dispose of the surface tension of the solvent, and this might influence the outcome of the simulation. To solve this problem, we will create now an approximately spherical droplet of MeOH before solvating the nanoparticle. The necessary program is **Excbox** (Section 5.5 of the main manual).

Excbox requires a *.bxi* file and a *.dat* file in MC-compatible format as input. Moreover, such files must have the same name. Start with the last frame of the equilibrated liquid you employed in part 1 of this exercise, *meohmc.bxo* and *meohmco.dat*. Copy them as follows and execute the program:

```
cp meohmc.bxo meohmc.bxi
cp meohmco.dat meohmc.dat
./run.excbox meohmc cut
```

Where “cut” is the prefix for output files. The program will ask for max centre of mass distance for solute and solvent: answer as follows.

```
give max com distance, solute and solvent
26.0 0.0
```

Excbox evaluates the centre of mass distance of every molecule in the simulation box from the cluster centre, and erases those that are farther than the cutoff you are specifying. 26.0 Å, in this case, is enough to cut the vertices of the cubic box without dramatically reducing the total molecule count (remember that you need enough molecules to solvate the nanoparticle). Note that you do not have “solvent” here, as only one species in the simulation box is automatically interpreted as “solute”. Now the program wants to know the number of slave cards for solute and solvent (1 card for the solute, the rotatable H-O-C-H torsion).

```
give number of slave cards, solute and solvent
1 0
original and final solutes and solvents 1024 885 0 0
please add the correct molecule count in heading of bxi and dat
files
normal end of operation
Thank you for using MiCMos
```

The calculation is made in a fraction of second. You can see that with this cutoff the number of MeOH molecules is now 885, which means a $\sim 14\%$ reduction. The program produced two output files, namely *cut.bxi* and *cut.dat*.

IMPORTANT: The new molecule count MUST be updated in the header of both files, as the program is not able to do this automatically.

Open the *cut.dat* file and look at the second line in the header, which should report the number of solute molecules (nmsolu), that of solvent molecules (nmsolv) and the two corresponding atom counts (nasolu, nasolv):

```
# 0 0 6 0 0
```

To

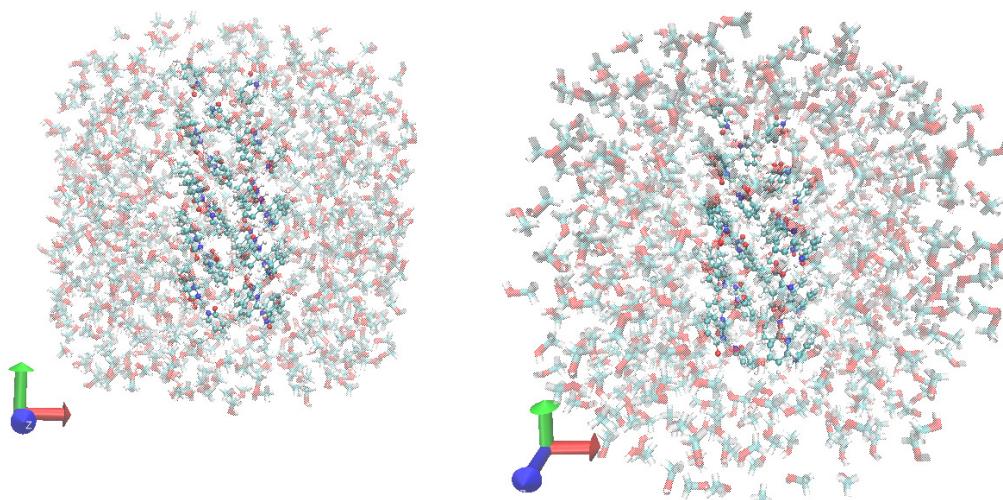
```
# 885 0 6 0 0
```

i.e. add the correct number of “solute” molecules (nmsolu) that survived the *Excbox* procedure. If solvent was present, you should have updated also the solvent count (nmsolv). In this exercise you do not need the *cut.bxi* file, thus you can safely cancel it.

Now you can execute **Nanosolv** (see Part 3 of this exercise):

```
./run.nanosolv pyridcut cut
```

Again, answer 0.0 when prompted to give the atol parameter. The program will end by informing you that the merged box file, *pyridcut.solv.dat*, contains now 723 solvent molecules. The droplet you obtained is shown below, both in orthographic (left) and perspective (right) projections. The shape of the solvent has a more isotropic appearance, even though it is not a perfect sphere.



You can now use exactly the same input and topology prepared for the cubic nanodroplet to carry on the simulation. Good luck!