MiCMoS
Milano Chemistry Molecular Simulation
Description and User Manual

Tutorials

T11. Analyzing MC and MD trajectories

PURPOSE: In this tutorial, it is described how to employ service programs ***Geomet***, ***Analys***, ***Distrib***, ***Correl***, ***Redene***, ***Datgro***, ***Naverag*** and ***Debye*** programs to analyze MC and MD trajectories.

REFERENCE to main manual: Section 8 (service programs).

FILES: You can find all the files you need to run this tutorial here:
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T11/b3mdc.dat
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T11/py1mdc.dat
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T11/py2mdc.dat
https://sites.unimi.it/xtal_chem_group/images/MiCMoS_package/T11/pyrid.oeh
The b3mdc and py1mdc starting .*dat* trajectory files have been produced in Tutorials T9 and T10 for liquid benzene (T9) and pyridone (T10). A 100 ps-long trajectory for the same pyridone crystal, py2mdc.*dat*, is also given, together with a pyrid.*oeh* file that should correspond to the structure file you produced in T9. You will also need macros to run ***Geomet***, ***Analys***, ***Distrib***, ***Correl***, ***Redene***, ***Datgro***, ***Naverag*** and ***Debye***. Place the .*dat* files and the pertinent (Windows/Unix) macros into your working directory.

Very interesting physics is hidden in your trajectory files. You can extract relevant information by using MiCMoS service programs described in the main manual, Section 8. In this Tutorial, a brief overview is given to show you how it is possible to prepare graphical output and even animate your trajectories, with the aid of some other free software available on the web.

A first good practice is to perform geometric checks to make sure that no unusual molecular deformations occur during the simulation. The ***Geomet*** module needs .*oeh* and .*top* file, plus a valid trajecory (1 frame is enough) .*dat.* Let's now perform the analysis on liquid benzene. The program is invoked by the command

```
./run.geomet benz benz b3mdc
```

using benz.*oeh*, benz.*top* and the b3mdc.*dat* trajectory. Output is in b3mdcgeo.*pri*. The program asks for:

```
torsion angle distribution
   give tau min, max,no. of bins max 300
```

The torsion angle spread is analyzed giving -50 to +50° range (TMIN, TMAX) with 21 steps (NTBIN); see manual, Section 8.1. Top oscillations are < 15° both for C-C torsion and for improper torsion.

```
-50 50 21
 distances and angles check
   give d-r and d-theta limit values for printout
```

Excess train is analyzed by answering with 0.7 Å for stretch deviations and 7° for bending deviations:

```
 0.7 7
 give step for n. of frame analysis
```

Answer 1 to check all frames or, say, 10 to perform the analysis every 10 moves (shorter output). For example, in this last case the program signals:

```
10
 frame n.      1  moves          0
 frame n.     10  moves       1800
 frame n.     20  moves       3800
 frame n.     30  moves       5800
 frame n.     40  moves       7800
 frame n.     50  moves       9800
 normal end of operation
 Thank you for using MiCMoS
```

CAUTION: problems may arise in singular situations (e.g. all torsion angles very close to zero, as in very flat molecules) if some particular bins are not included in the histogram of torsion angles. In this case, for example, answering -50 50 20 to the first question makes the program skipping the angle 0, which is essentially the unique populated in the starting frame. Therefore, ***Geomet*** stops with a warning message:

```
frame n.      1  moves          0
 torsion angle n., molecule n.     1     8
 torsion angle out of range, try extend by +- 5°
```

To avoid this problem, you can either erase from the trajectory the offending frame (the first one) or change the step by using an odd number for NTBIN. If this does not work, you have to play with input parameters until you find the correct combination.

Open the b3mdcgeo.*pri* output. After recalling what are the dihedrals in the .*top* file, the cell and the atomic coordinates in the .*oeh* file, the program prints the distribution of torsion angles.

The first column in the table shows the angle bins; the subsequent columns represent one torsion angle each, in the same order as specified at the beginning of the output (and corresponding to that in the .*top* file). From the example above, we can see that the first torsion (C6–C2–C1–C3) is roughly evenly distributed between –15º and +15º after 1800 MD moves.

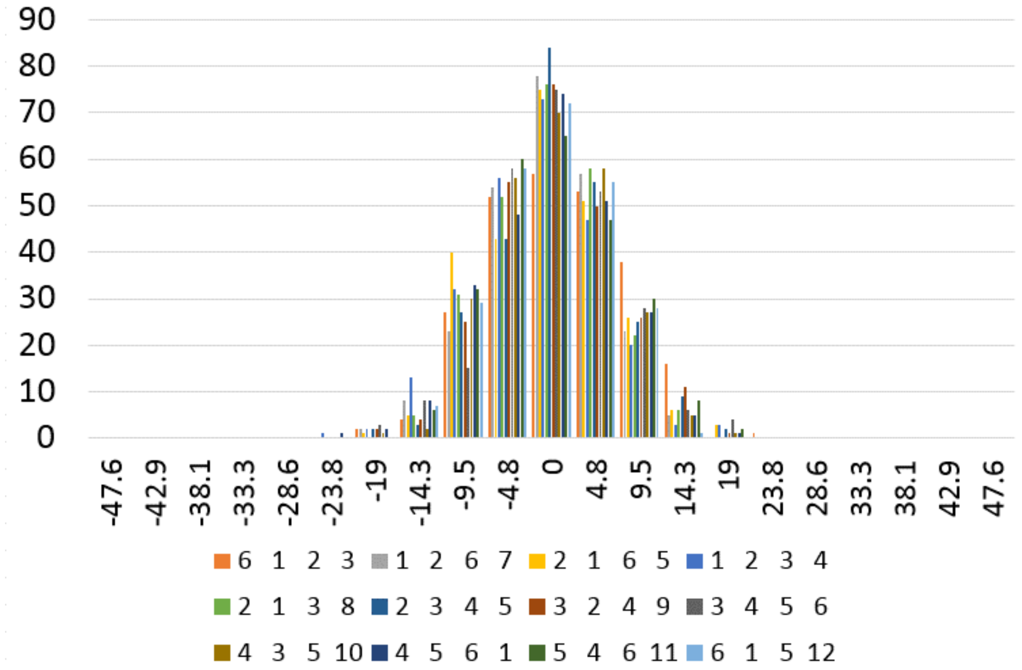
```
  +++++++++  MiCMoS 2.2 jul 2023  +++++++++
          This is the Geomet module

 reading from top file    #BENZNE99  benzene topology
 solute dihedral angle distribution, atoms   6   1   2   3
 solute dihedral angle distribution, atoms   1   2   6   7
....
 solute dihedral angle distribution, atoms   5   4   6  11
 solute dihedral angle distribution, atoms   6   1   5  12
   oeh file   #BENZNE99  benzene
      1.0000    1.0000    1.0000    90.000    90.000    90.000
    0.0000    1.3900    0.0000      0.0000    1.3900    0.0000
    1.2038    0.6950    0.0000      1.2038    0.6950    0.0000
....
   -2.1391    1.2350   -0.0000     -2.1391    1.2350   -0.0000
....
frame n.    10  moves       1800
  -47.6    0    0    0    0    0    0    0    0    0    0    0    0
  -42.9    0    0    0    0    0    0    0    0    0    0    0    0
  -38.1    0    0    0    0    0    0    0    0    0    0    0    0
  -33.3    0    0    0    0    0    0    0    0    0    0    0    0
  -28.6    0    0    0    0    0    0    0    0    0    0    0    0
  -23.8    0    0    0    1    0    0    0    0    0    1    0    0
  -19.0    2    2    1    2    0    2    2    3    1    2    0    0
  -14.3    4    8    5   13    5    3    4    8    2    8    6    7
   -9.5   27   23   40   32   31   27   25   15   30   33   32   29
   -4.8   52   54   43   56   52   43   55   58   56   48   60   58
    0.0   57   78   75   73   76   84   76   75   70   74   65   72
    4.8   53   57   51   47   58   55   50   53   58   51   47   55
    9.5   38   23   26   20   22   25   26   28   27   27   30   28
   14.3   16    5    6    3    6    9   11    6    5    5    8    1
   19.0    0    0    3    3    0    2    1    4    1    1    2    0
   23.8    1    0    0    0    0    0    0    0    0    0    0    0
   28.6    0    0    0    0    0    0    0    0    0    0    0    0
   33.3    0    0    0    0    0    0    0    0    0    0    0    0
   38.1    0    0    0    0    0    0    0    0    0    0    0    0
   42.9    0    0    0    0    0    0    0    0    0    0    0    0
   47.6    0    0    0    0    0    0    0    0    0    0    0    0
```

Then, the program prints all the bond distances and angles whose stretching and bending have implied a deformation larger than that specified as input (0.7 Å for stretch deviations and 7° for bending). The number of cases exceeding these thresholds is limited and the standard deviations of bond lengths and bond angles are acceptably small.

```
 large bend mol,i,j,k,thezer,dev    1    1    6   12 120.00   7.65
 large bend mol,i,j,k,thezer,dev    1    3    4   10 120.00   9.49
 .......
 large bend mol,i,j,k,thezer,dev  239    5    4   10 120.00   7.04
 large bend mol,i,j,k,thezer,dev  239    5    6   12 120.00  -8.15
 large bend mol,i,j,k,thezer,dev  250    6    1    7 120.00   7.70
n .molecules, n.stretch, total,n.devs, rmsd   250    12  3000     0   0.02337
 n.molecules, n.bends, total, n.devs, rmsd   250    18  4500   105      3.01
 center of mass coordinates      -0.13      0.00     -0.00
```

The output ends by summarizing relevant statistics on average stretching and bending deviations, plus information on their estimated standard deviations and overall centre of mass drift. We note that only 2 % of bending degrees of freedom (d.o.f.) have deviated by more than 7º (105/4500), with an overall estimated standard deviation as low as 3º. No relevant deviations in bond lengths have been found. Overall, the overall quality of the simulation is acceptable from the viewpoint of geometrical parameters.

The graphical output for distributions of torsion angles can be obtained after importing data into the Microsoft Windows® EXCEL environment:



Obviously, to make the output as much as possible informative, it would be better to select just a subset of torsions.

For liquids, the ***Correl*** module calculates the translational root mean square deviation (rmsd) and the rotational correlation function *C*(**u**,**u**) using the structure trajectory file. See manual, Section 8.4 for more information. This program is invoked by the command

```
./run.correl b3mdc
```

The program prompts you to

```
give n.of reference frame,tot n.of frames,timestep
```

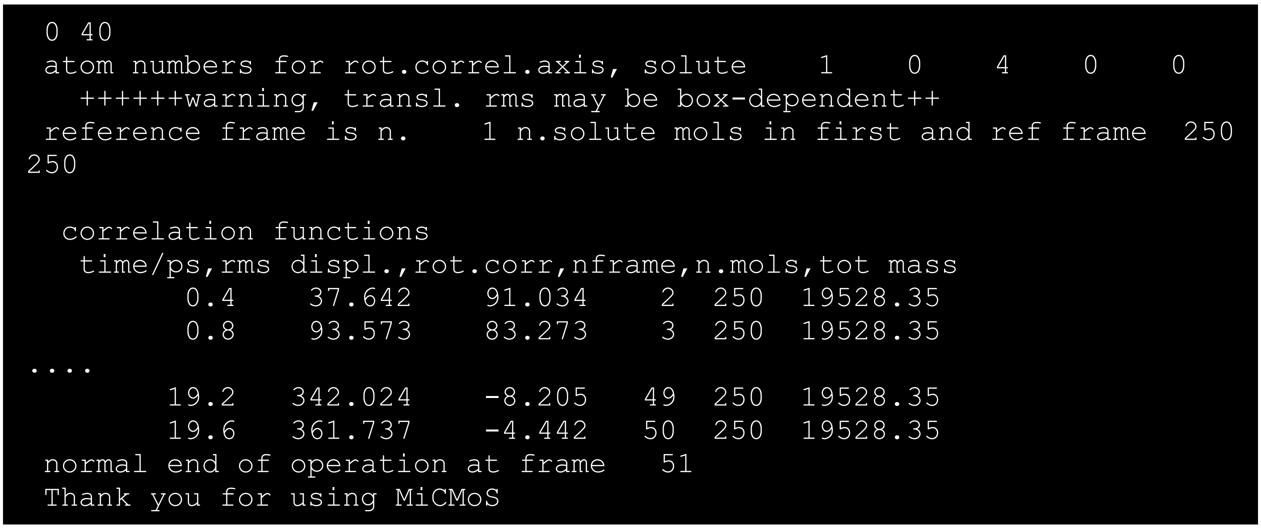
The analysis starts from frame 1 up to frame 50 (10000 steps with frames written every 200 steps, with Δt = 2 fs) in the output b3mdc.*dat* file. Answer 1 50 0.002.

```
1 50 0.002
 give ns. of 5 indicators for rotn. axis,solute
```

To define the angle for rotational correlation type 1 0 4 0 0 so the program will use the transannular C1-C4 vector (see Figure 8.1 in the manual).

```
1  0  4  0  0
  give atom code for rmsDr (0=c.o.m.) and max dist
```

For the translation rmsd, use the center of mass (0) and a distance limit of 40 Å. See manual, Section 8.4 for more information.



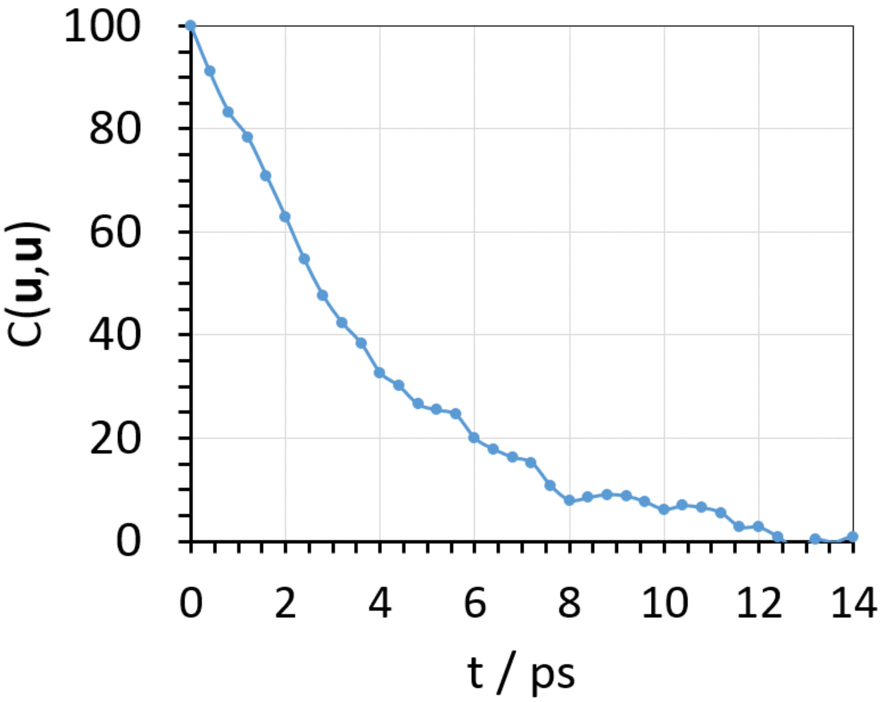
```
 0 40
 atom numbers for rot.correl.axis, solute     1     0     4     0     0
   ++++++warning, transl. rms may be box-dependent++
 reference frame is n.     1 n.solute mols in first and ref frame   250
250

  correlation functions
   time/ps,rms displ.,rot.corr,nframe,n.mols,tot mass
         0.4     37.642     91.034     2   250   19528.35
         0.8     93.573     83.273     3   250   19528.35
....
        19.2    342.024     -8.205    49   250   19528.35
        19.6    361.737     -4.442    50   250   19528.35
 normal end of operation at frame    51
 Thank you for using MiCMoS
```

Output is in b3mdccor.*pri*. Relevant columns are the first (time of the simulation step, in ps), the second (RMS displacement, equation 8.7 in the manual) and the third (rotational correlationfunction, equation 8.6 in the manual) ones.

CAUTION: The translational correlation is affected by discontinuities caused by the possible displacement of entire molecule through one or more box dimensions, occurring during the simulation to prevent molecules from leaving the box; hence it is unreliable. The rotational correlation function correctly shows a loss of rotational coherence within a few ps.

If you export the ASCII output of b3mdccor.*pri* in a graphical commercial software like EXCEL, you can tralslate the tabular output into a graphical one:

Liquid benzene, MD rotational correlation function.



The ***Analys*** module performs a variety of tasks as described in the manual, Section 8.2. If ***Geomet*** is mainly devoted to intramolecular geometry checks, ***Analys*** studies intermolecular contacts and provides us with centre–of–mass radial distribution functions (RDF’s), as well as selected atom–atom RDF’. It is invoked by the command:

```
./run.analys benz b3mdc
```

Here “benz” is the prefix of benz.*top* and b3mdc that of the MD trajectory. The program asks for the following parameters on the fly:

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

It is a good idea to limit the analysis to regions of the trajectory corresponding to a fully equilibrated system. This is not exactly the case for this simple tutorial; therefore, as an illustrative example of how the program works, we will use just the last 10 frames, where we are sure that the liquid is fully randomized as molecules have completely lost their correlation (see above). Thus, answer 30 50.

```
30 50

  check short intermolecular distances
 give % of contact R, e.g.90., zero to skip
```

Answer 90: print intermolecular distances that are < 90 % of sum of van der Waals atomic radii.

```
90.0
 g(R) calculations, 0=yes, 1=no
```

We want to see some RDF’s, so type 0.

```
0
give pack coeff if box not present,0.5liq-0.7cry
```

The packing coefficient (cpack) is required at this stage, but it is used only for non-periodic systems (clusters, droplets). As we are dealing with a crystal with full periodicity, we can answer 0.0 and go on.

```
0.0
c.o.m. radial distribution function
 give rmin,step,n.bins, zeros to skip
```

Suggested parameters for the centre–of–mass distribution (c.o.m) RDF are: 2. 0.2 70 0. See the manual (Section 8.2) for detailed explanations. Briefly, the RDF will start from 2 Å, and it will consist of 70 steps each 0.2 Å wide. Thus, the graph will end at 2+70·0.2 Å = 16 Å. The last entry can be safely left 0 (see the manual).

```
2.  0.2  70  0.


 atom-atom RDFs,INTERMOLECULAR ONLY
 give rmin,step,n.bins max 300,zeros to skip
```

For the moment, we don’t want atomic RDF’s, as benzene has neither relevant nor structure determining atom–atom interactions, so set all values to 0 (0 0 0).

The program will scan the requested frames (30–50) and for each of them will print a list of short atom–atom interactions, followed by the RDF distribution in a tabular form. Have a look at the b3mdcanl.*pri* output file:

```
  +++++++++  MiCMoS 2.2 jul 2023  +++++++++
           This is the New-Analys module

 dat file is trajectory at MD st solutes and solvents   250    12     0     0
 frame n.    30  moves       5800
 short inter, at-mol-types    1    1   12    5   50   12  3.04  3.54
 short inter, at-mol-types    4    1   12    5   51   12  3.16  3.54
....
 short inter, at-mol-types    4  244   12    9  246    2  2.41  2.87
 short inter, at-mol-types    5  244   12   12  245    2  2.50  2.87

 solute-solute intermol.dist,given limit,min val
      2   12      2.583        85     2.377
     12   12      3.186        77     2.739
 c.o.m. radial functions,unsmoothed-smoothed
   solute-solute, solvent-solvent,solute-solvent
    2.100  0.000  0.000
    2.300  0.000  0.000
....
   15.700  1.002  1.016
   15.900  1.032  1.028
--
```

As for the short distances, lines beginning with “`short inter, at-mol-types`” contain: number of the atom in the first molecule (same order as in the .*top* file), number of the first molecule (same order as in the .*dat* file), atomic species code number (as in Table 1.1), number of the atom in the second molecule (same order as in the .*top* file), number of the second molecule (same order as in the .*dat* file), atomic species code number (as in Table 1.1), actual distance (Å), van der Waals contact distance (Å). Finally, total number of contacts are summarized: for example, we have 85 H···C short contacts with minimum distance of 2.377 Å and 77 C···C ones with minimum distance of 2.739 Å. They are both significantly shorter than the corresponding sums of van der Waals radii (2.583 Å and 3.186 Å).

The output (b3mdcanl.*pri*) reveals quite a number of short contacts, as is appropriate for the liquid state where rotational and translational diffusion causes continuing clashes.

Note that the RDF consists of two “y” columns: the second one is relevant, as it has been smoothed by an ad–hoc algorithm.

The analysis of the MD/MC output should be made on an average of the last frames of the MD/MC run: this can be obtained using module ***Distrib*** (manual, Section 8.3). the running command is:

```
./run.distrib b3mdc
```

where "b3mdc" is the prefix of the b3mdcanl.*pri* file produced by ***Analys***. ***Distrib*** recognizes that the ***Analys*** output contains more than one distribution, and provides an average one, together with point–by–point estimates of the corresponding standard deviations.

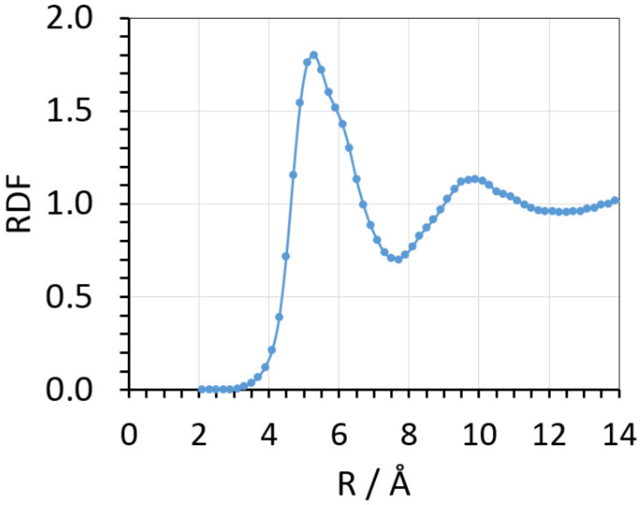
```
Analyzing frame n.    30, move n.       5800
Analyzing frame n.    31, move n.       6000
....
Analyzing frame n.    49, move n.       9600
Analyzing frame n.    50, move n.       9800
Averaged    21  frames
 Job executed regularly
 Have a nice day
```

The printfile is b3mdcdistr.*pri*. Let's open it:

```
  +++++++++  MiCMoS 2.2 jul 2023  +++++++++
          This is the Distr module

       Distr module version 1.0-2020

Average COM RDF from    21  frames: solute-solute
     R / A       RDF       ESD
     2.100     0.000     0.000
     2.300     0.000     0.000
....
    15.700     0.995     0.012
    15.900     1.013     0.008
```

You may now plot this distribution with EXCEL:

Liquid benzene MD, centre–of–mass RDF from the last 21 frames of the simulation



RDF > 1 corresponds to centre–of–mass distances that are significantly more frequent than in an uniform and totally random distribution, mirroring relevant structural aspects of packing. The present c.o.m. RDF shows a broad peak near 5 Å, as observed experimentally.

We now switch to a crystal case (pyridone, whose simulation has carried out in Tutorial T10). Proceed with ***Analys*** as above; the running command is now

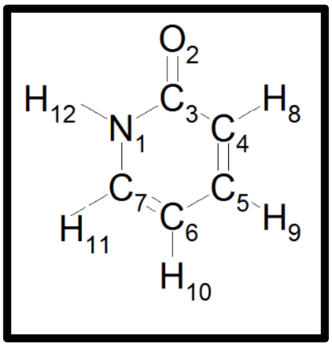
```
./run.analys pyrid py1mdc
```

You must have both the py1mdc.*dat* trajectory and the pyrid.*top* file in your working directory. When prompted, use 2.0 0.1 100 0.0 for `rmin,step,n.bins, cpack for c.o.m.` RDF. When it comes to atom–atom RDF's, do not type 0 and confirm the same parameters:

```
atom-atom RDFs,INTERMOLECULAR ONLY
 give rmin,step,n.bins max 300,zeros to skip
2.0 0.1 100
```

Now the program will want *ncpu*, the number of specie pairs for the "solute". Individual RDF will be computed for each of the ncpu pairs of atomic species.

```
give ncpu,number of solute rdfs, 0 skip, -1=do all
```



Be careful: Computing the RDFs for all the atom–atom pair types is **very time–consuming**, especially when large or complex molecules are studied. In this case, we will focus just on atoms that could be reveal interesting properties of the hydrogen bond network in crystalline pyridone. See the scheme on the left: they are H12 (atom specie id 7, Table 1.1 in the main manual) and O2 (atom id specie 27, Table 1.1 in the main manual). Thus, it is wise to focus just on contacts between atomic species 7 and 27. Answer 1 to the former request and proceed by typing in the desired specie id numbers:

```
1
 now give ncpu pairs of atom types ***i-j with i<j***
7 27
give ncpv,number of solvent rdfs  0 skip -1 do all
```

We have no solvent, so we can answer 0 to both this request and the next one:

```
0
 give ncpuv,n. of solute-solvent rdfs,0 skip -1 all
0
radial function smoothing, 0=yes, 1=no
```

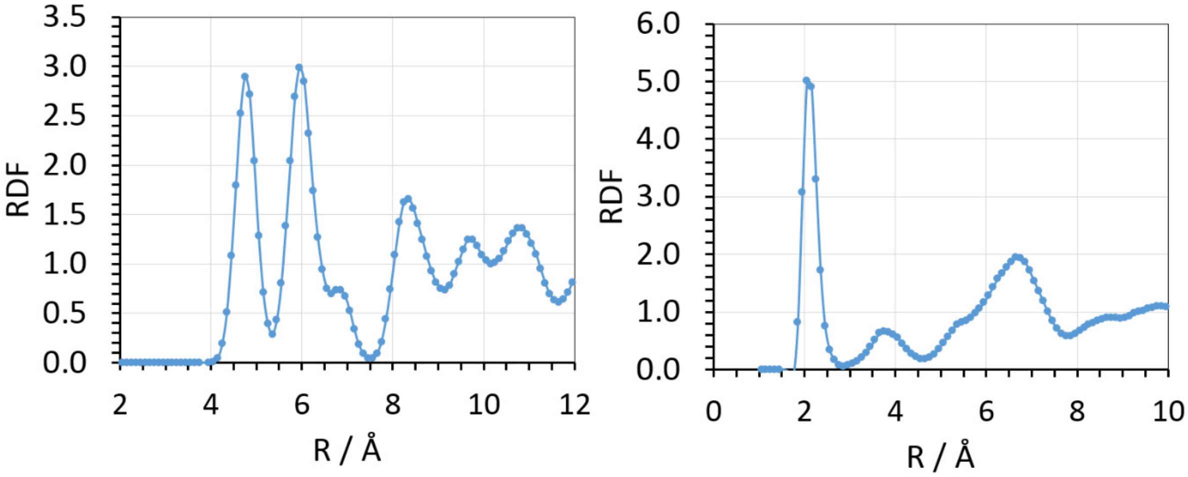
We want to deal with smoothed functions to avoid unnecessary noise. Answer 0 again. The program will proceed and will print a py1mdcanl.*pri* output in a few seconds. It contains both c.o.m. and NH···O intermolecular RDF's. Call ***Distrib*** to average them:

```
./run.distrib py1mdc
```

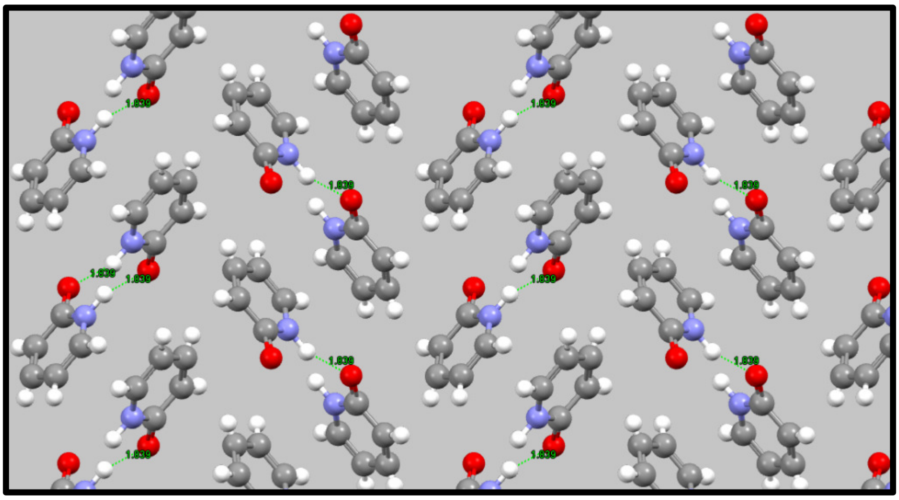
and transfer the average distributions in the tabular py1mdcdistr.*pri* output to EXCEL for plotting (graphics in the next page). The atom-atom RDF mirrors the set up of a strong H···O hydrogen bond, with a very high and sharp peak around 2.0 Å. together with the c.o.m. RDF, we can conclude that packing of pyridone is dominated by strong attractive interactions, likely setting up extended hydrogen–bonded networks. This is confirmed by opening the starting pyrid.*cif* file with the CCDC Mercury program (next page). The amide group forms very short

NH···O hydrogen bonds with itself, setting up infinite *zig–zag* ribbons orthogonal to the (*a*,*b*) plane.

Pyridone crystal. Left: c.o.m. RDF. Right: H12···O2 RDF.



Crystal packing of pyridone, as retrieved from CSD (file pyrid.cif) and viewed along the c axis. Molecules form zig–zag ribbons perpendicular to the (a,b) plane. Hydrogen bonds are highlighted in green. The experimental NH···O distance is 1.84 Å, a bit shorter than the MD estimate.



Structural information should be complemented with energetic data whenever is possible. The ***Redene*** module (manual, Section 8.5) plots energies and box dimensions as a function of simulation time, and calculates the averages and root mean square deviation (rmsd) of the corresponding values over the selected time window.

CAUTION: Ensure that *nwre* parameter is active in the *.mdi* command file when you run your dynamics (see manual, Section 7.6.2); otherwise, no *.ene* output file will be written for the subsequent analysis with ***Redene***.

The program must read a *.ene* file produced by Monte Carlo or Molecular Dynamics modules. See manual, Section 8.5.1 for an exhaustive explanation of all entries in the *.ene* file.

The command is:

```
./run.redene py1md
```

using the py1md.*ene* output file. Output is in py1mdene.*pri*. Answer the dialog mode, which asks for:

```
give multiples of cell dimens. (1,1,1 or 0,0,0 ok)
```

These are necessary to normalize total energies to crystallographic unit cell. For liquids, repetitions are 1 1 1; for crystals, these are the factors you used to generate the simulation box with Boxcry (see manual, Section 5.1 and Tutorials T8 and T10). You can recall them directly from the trajectory .dat file, as they are the last entry in the third line of the eading of each frame. Otherwise, they are also printed both at the beginning (line header "`unit cell parameters`") and at the end (line header "`final box edges, multiplicities`") of the MD .output file. As for the present case, they are 2 5 5:

```
2 5 5
 give MD timestep (=1 if MC)
```

Timestep is 0.002 ps. In Monte Carlo, this entry should have been 1, as suggested, as time is not a dynamical parameter.

```
0.002
 nmol,natom,solute-solvent  200   12    0    0 mol. weights   95.10    0.00
 no. of energy and box data, last move     52   10000.0
 give min and max move/time for averages,zero to skip
```

The user can choose to discard the first part of the simulation when the system is not yet in a steady state, by typing in the desired time window; in this case 10. to 20. ps should be adequate.

```
10. 20.
 normal end of operation
 Thank you for using MiCMoS
```

The program prints the output in a py1mdene.*pri* file. Open it:

```
starting energies  -7773.0  -7893.6     33.9      0.0 -15632.7
 starting partitioned energies  -7773.0  -7893.6      0.0      0.0      0.0      0.0
 starting box   27.3140   29.5600   28.5050     90.00     90.00     90.00       1.4
 box multiplicities   2.0   5.0   5.0
       0.     -38.9     -39.5       0.2       0.0     -78.2     -78.3       0.0       0.0
       0.     -48.0     -34.3      22.8       0.0     -59.5     -82.4       0.0       0.0
.....
       0.       0.0       0.1       0.1       0.0       0.0       0.0       0.0       0.0
       0.      10.3       6.6       6.0       0.0       0.0       0.0       0.0       0.0
       1.      11.1       8.7       6.5       0.0       0.0       0.0       0.0       0.0
.....
      20.     1.352    13.233     5.929     5.956      90.0      90.0      90.0
      20.     1.352    13.233     5.929     5.956      90.0      90.0      90.0

   averages,min and max move,n.points      10.0      20.0      27.0

   aver.and sigmas of elp,eq,eintru,eintrv,etot
   -46.4     0.3   -32.8     0.2    35.9     0.7     0.0     0.0   -43.2     0.7

   averages and sigmas of euu,evv,euv
   -79.1     0.2     0.0     0.0     0.0     0.0

   averages and sigmas of a,b,c,alf,bet,gam,dens
  13.321    0.088    5.910    0.012    5.924    0.030   90.000    0.000   90.000    0.000   90.000    0.000    1.355
0.004
 normal end of operation
```

Redene starts giving information on energies and box, and then prints their evolution – as well as density – as a function of the step number. In the first output block, are shown the simulation time (in ps), Lennard–Jones energies, Coulomb energies, intramolecular energies for "solute" and "solvent" (zero here), and total energies. In the second one, you may find total intramolecular potentials, i.e. stretching, bending and torsion contributions. In the third block, interactions energies with the barriers are given; if you are not dealing with a confined simulation, this block is composed of a long sequence of zeroes. In the fourth block, crystal density and cell edges are reported. The output ends with statistics: min and max move, number of points, average energies and estimated standard deviations, followed by average cell edges and angles. The latter are here constrained to 90.0º by symmetry in pyridone: no Parrinello–Rahman anisotropic pressure scaling has been activated (see manual, Section 7.3.3).

Further visual inspection tools can be provided by the free VMD (Visual Molecular Dynamics) program, from the NIH Center for Macromolecular Modeling & Bioinformatics, Theoretical and Computational Biophysics Group, University of Illinois, USA (https://www.ks.uiuc.edu/Research/vmd/). Unfortunately, it cannot interpret directly MiCMoS *.dat* trajectory files: they should be first converted into GROMOS–compatible *.gro* files by the program ***Datgro***. The command is:

```
./run.datgro py1mdc
```

where py1mdc is supposed to be the prefix of the trajectory *.dat* file to be converted. The program asks for:

```
Give initial and final frame number to be converted in your trajectory
```

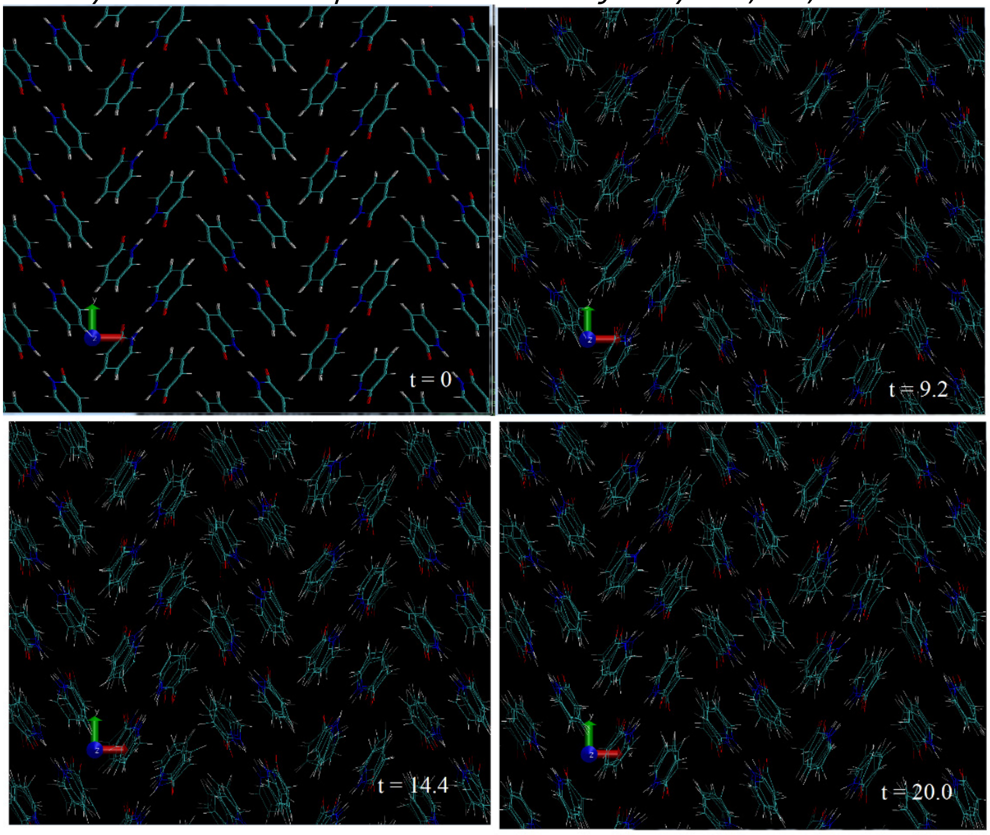
As the illustative trajectory of Tutorial T10 is very short, it can be converted entirely into *.gro* format. Be cautious, though: *.gro* file are bigger than *.dat* ones, so it is wise to select just the part of the trajectory in which you are truly interested (usually, that corresponding to the fully equilibrated system). For the moment, answer 1 51 to include all frames.

```
1 51
.gro file will be printed from frame     1  to frame    50
Start working with frame       1  and going on...
 Normal termination of the program
Skipped       0; written      51  frames
 Thanks for using MiCMoS. Have a nice day
```

Datgro has now produced a file called py1mdc.*gro*: this is the only output to be expected. If you open this file, you can check that first step is labelled 0 and the last one 10,000, as expected. Note that the program has assigned to all "solute" molecules an arbitrary "SOLU" label; "SOLV" would be the one for the "solvent", if present. Atoms are numbered in ascending order, following the same organization of the *.dat* file.

Now, input the py1mdc.*gro* file to VMD (or to any other GROMOS–compatible graphical software) and do whatever you need. Refer to the VMD documentation to learn how to deal with the program. For example, you can follow the trajectory evolution in time:

Pyridone crystal at 298 K. Snapshots of the MD trajectory at 0, 9.2, 14.4 and 20.0 ps.



It is not always straightforward to compare the output of the whole simulation box with experimental crystallographic results. In MiCMoS, you can use the module ***Naverag*** to produce a spacetime average crystallographic cell from the frames of your trajectory. The program also estimates atomic thermal ellipsoids by putting together the molecular internal vibrations with the centre of mass oscillations. To have meaningful results, you should ensure that te system is equilibrated, and that the trajectory is long enough to sample all the relevant atomic motions.

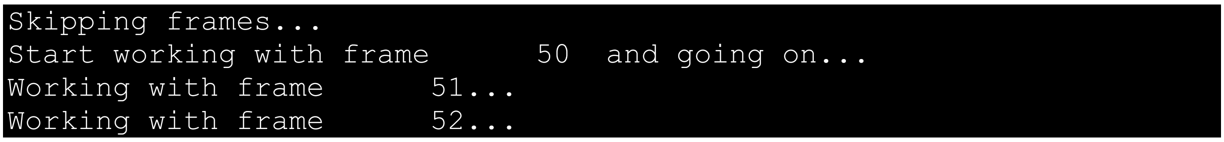
You can execute this program on the other trajectory of the pyridone crystal at 298 K (py2mdc.*dat*) that goes with this Tutorial. This trajectory is a bit longer and the results are thus more accurate. Please digit:

```
./run.naverag py2
```

The program will prompt to give the integration limits:

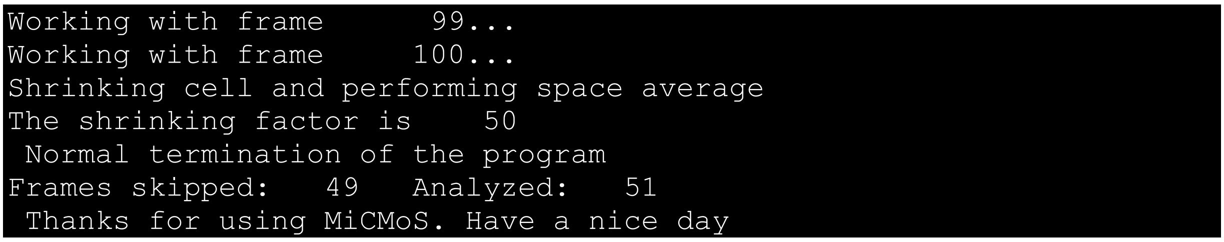
```
              +++++++++  MiCMoS  +++++++++
        Milano Chemistry Molecular Simulation
                 release 2.2 jul 2023

               This is the new averag program

      Give initial and final frame number to be analyzed
```

Answer, for example, 50 100. The program will inform you on which frame it is working:



```
Skipping frames...
Start working with frame       50   and going on...
Working with frame       51...
Working with frame       52...
```

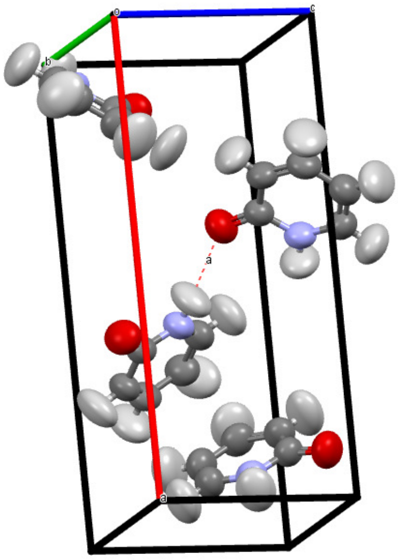
The job will be completed in a few seconds:



```
Working with frame       99...
Working with frame      100...
Shrinking cell and performing space average
The shrinking factor is     50
 Normal termination of the program
Frames skipped:    49   Analyzed:    51
 Thanks for using MiCMoS. Have a nice day
```

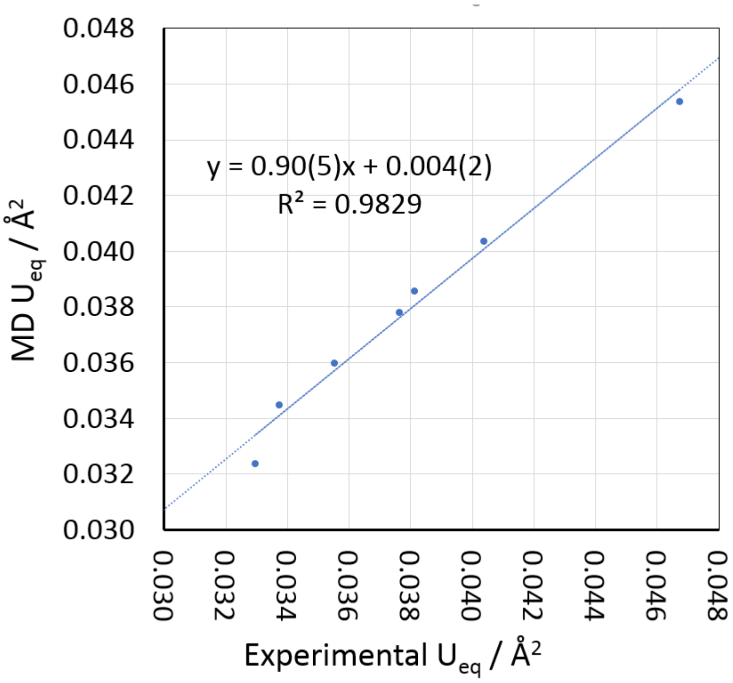
Naverag produces three output files. py2service.*out* contains information on the individual atoms in the time average simulation box and specifies on what molecules the final space average is done. Most important are the other two outputs, py2ave.*dat* and py2ave.*res*. py2ave.*dat*, which is a time average frame in standard MiCMoS .*dat* frame format. This can be visualized by VMD after converting it into a .gro file with ***Datgro*** (see above), and may be employed as a suitable input for any other calculation. For example, it can serve as a starting point for futher MD steps, or it can be analyzed with ***Geomet***, ***Analys***, ***Correl*** and ***Distrib*** to extract information on the time average structure. Finally, py2ave.*res* is a standard shelx file with the spacetime average crystallographic structure. From that, it is possible to visualize MD-derived average thermal ellipsoids. This pseudo-crystallographic cell in the .*res* file is always treated as P1, as no symmetry constraintes can be applied during the MD or MC simulations. It comes from the space average of the whole simulation box, which is shrunk by a NREPA x NREPB x NREPC factor (=50 in this case; see main Manual, Section 5.1).

MD-predicted average crystallographic strructure of pyridone at T = 298 K, with predicted Debye–Waller ellipsoids shown.



There are several ways to check quantitatively the quality of the prediction. A possibility is to compare the equivalent U’s of your simulated atoms with the experimental ones (for pyridone see Ohms *et al.*, Zeitschrift für Kristallographie 169, 185-200 (1984)). The MD estimates are roughly 2 times higher than the corresponding X-ray quantities: you must bring them on the same scale. You can do this by computing the average $(U_{eq})_{experimental}/(U_{eq})_{MD}$ ratio for non-H atoms, which is 0.439 in the present case. Then, you can multiply $(U_{eq})_{MD}$ values by this factor and compare the two datasets graphically, as shown below.

MD-predicted U_{eq} (Å^2) vs. experimental ones for non-H atoms in crystalline pyridone at room temperature. A 0.439 scaling factor was applied to computed values to bring them on the same scale as the X-ray experiment. The linear regression is also shown, with least-squares standard deviations of parameters in parentheses.



The ***Debye*** module can simulate the total scattering output from your simulation box by exploiting the Debye scattering equation (main Manual, Section 8.8). To save time, in this tutorial you will use just the last frame of the longer trajectory of pyridone (py2mdc.*dat*). Note that you could obtain the same results by reading the mdo.*dat* file produced by the MD program, ***mdmain***.

First, you should set up the input file, specifying the steering parameters that will tell ***Datgro*** what to do. Open an empty file called pyrid.*inp* and write the following instructions. You may cut-and-paste the example in the Manual (Section 8.8.1) and then edit it according to your needs.

```
# nstart nend (first and last frame number, 0 0 to analyze all)
    101   101
# lambda, theta min, theta max, step
   1.54     2.0       30.0      0.25
# ilp (=0 apply, =1 not apply LP correction), thm(theta monochrom.)
                  1                                  13.3
# ibox (0=aperiodic, 1=periodic), cutoff
                  0                0.0
# ihyd (0=include,1=exclude H)
                  0
# iprint (0=normal, 1=extended printout)
                  0
```

This prepares an aperiodic calculation (fast, only the simulation box will be considered), without the Lp correction (the thm parameter is irrelevant) but including all the hydrogen atoms.

Now call the program:

```
./run.debye pyrid pyrid py2mdc
```

The first "pyrid" string refers to the input (pyrid.*inp*) you just prepared; the second one calls the pyrid.*oeh* structural file, which is used only to read atom identities. Finally, py2mdc.*dat* is the trajectory file.

The program prints some control strings on screen:

```
****************************************************
                   MiCMoS v2.2
                  July, 31th 23
             This is the Debye module
 ****************************************************

   title of oeh file is #PYRIDO11 'P
 frames analyzed: from n  101  to n  101  total     1
 only 1 frame will be analyzed - no average will be done
 skipped frame         1
 skipped frame         2
...
```

As you want to read just the last frame, the first 100 ones will be skipped; then, atoms type codes (see main Manual, Table 1.1) are recovered from pyrid.*oeh* and compared with those in the frame .*dat* file:

```
 ...
 skipped frame        99
 skipped frame       100

  title of coordinate file is   trajectory at MD st
  box with molecules, atoms, total   200  12  2400
 check of atom types in oeh and dat
    1   N   21
    2   O   27
    3   C   10
    4   C   12
    5   C   12
    6   C   12
    7   C   12
    8   H    2
    9   H    2
   10   H    2
   11   H    2
   12   H    7
 working with frame       101...
 scattering factor calculation complete
```

If all works properly, in a few seconds the program will print:

```
end of frame stream, found          1  frames
 finalize output
 Normal end of program
 Thank you for using MiCMoS
```

Open the file py2mdcdeb.*pri*: after a summary of the input parameters, you will find a table with 2θ, Q = 4π sinθ/λ and the corresponding scattering intensity:

```
   title of oeh file is #PYRIDO11 'P
 frames analyzed: from n  101  to n  101  total    1
 only 1 frame will be analyzed - no average will be done
 lambda (A),theta min and max,step(deg)    1.5400      2.00     30.00      0.25
 no LP correction applied
 Calculation on an isolated (non-periodic) frame
 hydrogens included in the calculation
 normal printout

  title of coordinate file is   trajectory at MD st
  box with molecules, atoms, total   200  12  2400


 average output from Debye scattering equation

   2th/deg    Q /A-1   Intensity
     4.000     0.285    3440.7
     4.500     0.320    3776.6
     5.000     0.356    2356.8
     5.500     0.392     903.3
     6.000     0.427     292.2
     6.500     0.463     315.4
     7.000     0.498     463.0
     7.500     0.534     463.1
...
    59.000     4.018      34.4
    59.500     4.049      32.6
    60.000     4.080      31.4
 Normal end of program
 Thank you for using MiCMoS
```

Note that the other output file, py2mdcprof.*pri*, contains exactly the same information, as no average was performed in this calculation. If more than 1 frame is included in the calculation, the prof.*pri* file contains the individual I(2θ) scattering curves of all the frames analyzed, while the deb.*pri* one prints the average I(2θ) one. Now rename the py2mdcdeb.*pri* file into py2mdcdeb_1.*pri* to save the output. Open again the pyrid.inp file and change it according to the instructions highlighted in green:

```
# nstart nend (first and last frame number, 0 0 to analyze all)
    101   101
# lambda, theta min, theta max, step
   1.54     2.0       30.0      0.25
# ilp (=0 apply, =1 not apply LP correction), thm(theta monochrom.)
                  1                                  13.3
# ibox (0=aperiodic, 1=periodic), cutoff
                  1               25.0
# ihyd (0=include,1=exclude H)
                  0
# iprint (0=normal, 1=extended printout)
                  0
```

The only difference is that the structural model is now periodic, with cutoff 25 Å. This means that twenty-seven replicas of the original simulation box will be built around it, to dispose of truncation effects. The Debye scattering equation will be applied to such an approximate bulk structure, including molecules up to 25 Å apart from the centre of coordinates of the starting simulation frame.

Save the .*inp* file and call again ***Debye***:

```
./run.debye pyrid pyrid py2mdc
```

If still present in your work directory, previous output files will be overwritten. Note that this time you have to wait a bit longer. Eventually, an output analogue to that commented before is produced. Save it, for example by renaming py2mdcdeb.*pri* into py2mdcdeb_2.*pri*, and open once more the input file. This time, raise the cutoff parameter to 30.0 Å. Leave the rest of the file unchanged:

```
# ibox (0=aperiodic, 1=periodic), cutoff
                   1               30.0
```

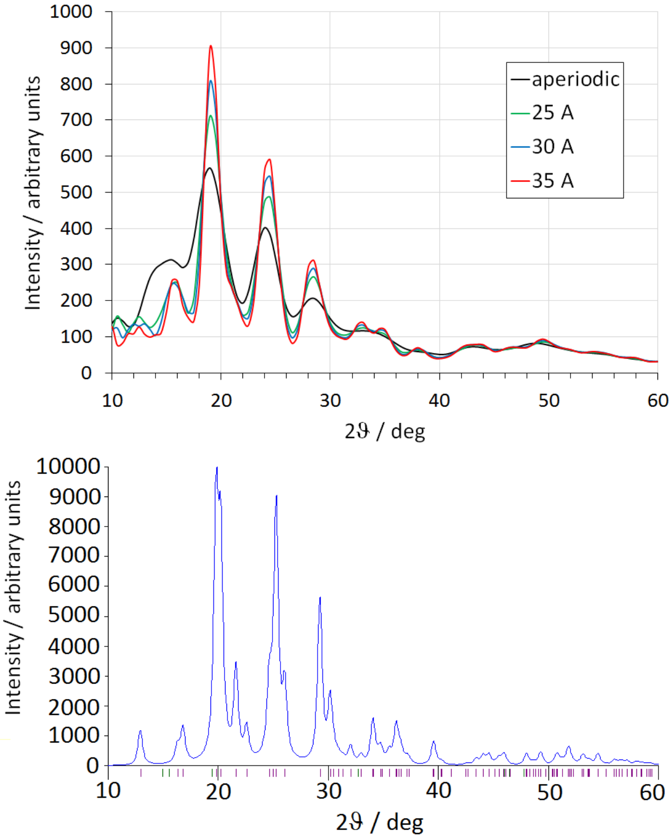
Repeat the calculation, which will be a bit slower (but should be completed in a few minutes, depending on the velocity of your processors). Save the py2mdcdeb.*pri* file (for example, call it py2mdcdeb_3.*pri*) and raise cutoff again (to 35 Å) in the input file:

```
# ibox (0=aperiodic, 1=periodic), cutoff
                   1               35.0
```

Repeat the calculation for the last time. It should last 5-10 minutes on a fast machine. Finally, put together the output of your three simulation by plotting them, for example, with Windows® EXCEL. The result is shown on the next page. The total scattering output is compared with the X-ray powder diffractogram predicted by Mercury® from the deposited PYRIDO11.*cif* file.

The resolution is obviously still rather low, but the main placement of the peaks and the intensity ratios are essentially correct, although the profile is not corrected for Lorentz-polarization. The peak width decreases, and peak height increases with increasing number of atoms in the sample. Ignoring periodic boundary conditions mainly implies that Fourier-truncation features appear at low angle (below 10 deg, not show). Increasing the cutoff distance makes more and more evident the details of the diffraction output.

Above: total scattering output, as predicted by the Debye scattering equation, for the MD model of the pyridone crystal simulated for 100 ps at RT. Only the last frame of the trajectory was considered (dimensions: ~ 28 x 31 x 30 Å). The black curve comes from an isolated simulation box; the other ones were obtained by exploiting peridicity, at increasing cutoff diatances from the centre of coordinates (green: 25, blue: 30; red: 35 Å).
Below: predicted X-ray diffraction output from the fully symmetric crystallographic structure of pyridone (CSD refcode: PYRIDO11).


Intensity / arbitrary units
2ϑ / deg
aperiodic
25 A
30 A
35 A
Intensity / arbitrary units
2ϑ / deg