**MDANSE 2018 SCHOOL TUTORIAL**

**Setting up a classical MD simulation with DL_POLY**

(Miguel A. Gonzalez, September 2018)

## 1. Introduction

In this tutorial you will learn how to prepare and simulate some simple systems (molecular liquids or molecular crystals), using some of the tools that are freely available for academic studies. The choice is based only on my personal preference, as DL_POLY is the MD simulation package that I know best. It is a very flexible MD engine, providing a large set of different potential functions and giving the user much freedom to simulate a large variety of systems. The manual is very clear and informative, and the input required by the program is well structured and relatively simple to understand and to prepare. In many cases (at least, for simple systems) this can be done manually by the user, with just a moderate effort. For large molecules, this work can become more cumbersome, but the utility program DL_FIELD can help enormously with this task.

However, this is only one among many other possibilities. For example, people interested in biomolecular systems could be best served by Gromacs or NAMD/VMD. And LAMMPS can provide even more flexibility and additional capacity. So if you are already familiar with some other software, you can either follow this tutorial to learn about DL_POLY or try to run the same examples using the tools you are used to. In any case, by the end of the session, any feedback about the advantages or disadvantages of different tools would be highly appreciated and should be shared with the other participants. You could even consider writing an alternative tutorial proposing other tools, which could be very helpful for some of your colleagues or for future editions of this school.

## 2. **Tools used**

*DL_POLY classic* (dl_class_1.10)[1]: MD engine.

*DL_FIELD* (dl_field_4.3)[2]: Utility tool to facilitate the creation of the input files needed by DL_POLY.

*Avogadro*[3]: Molecular editor and visualizer.

Open Babel[4]: Toolbox to convert between chemical file formats.

Aten[5]: Molecular modelling package that allows editing and manipulating atoms and molecules in isolated and periodic systems. Not explained in this tutorial, but it can be a very useful tool to work with DL_POLY. Usage examples are found in https://www.projectaten.com/aten/docs/examples, so you could try to do some of them to familiarize with the software and then use it to redo the exercises here.

Mercury[6]: Program for crystal structure visualization, exploration and analysis.

[1] Freely available from https://ccpforge.cse.rl.ac.uk/gf/project/dl_poly_classic/, but project is frozen. Preferable to move to the actively developed project DL_POLY_4 (http://www.ccp5.ac.uk/DL_POLY/).

[2] Freely available under an academic licence: http://www.ccp5.ac.uk/DL_FIELD.

[3] Free, open source code available at https://avogadro.cc/.

[4] Free, collaborative project available at http://openbabel.org/wiki/Main_Page.

[5] Free molecular modelling package available at https://www.projectaten.com/.

[6] Freely available from https://www.ccdc.cam.ac.uk/support-and-resources/Downloads/, although some features require a valid CSD license.

**3. DL_POLY**

DL_POLY requires three files that must be compulsorily named CONFIG, CONTROL, and FIELD.

- *CONFIG*: Contains the dimensions of the unit cell, a key to indicate the type of boundary conditions applied, and the coordinates of each atom (and eventually also velocities and forces).
- *FIELD*: Contains all the force field information defining the system to be simulated and the nature of the interatomic molecular forces.
- *CONTROL*: Contains all the directives controlling the simulation.

The program does not really have any UI allowing to prepare easily those files, so you will need to manage by yourself to create the three files following the needed specifications. In the program distribution, there is a Java interface that could be used as a starting point to prepare such files and also contains some analysis tools to analyze the trajectories generated by DL_POLY, but the possibilities offered by this interface are somewhat limited. A very interesting alternative seems to be the Aten project (https://www.projectaten.com/) by Tristan Youngs (ISIS, STFC, UK).

**4. Case 1: Liquid ethanol**

The file ethanol.xyz contains the structure of a single ethanol molecule. You can launch Avogadro to load it and visualize the molecule and check the geometry (e.g. measure bond distances and angles).

Now copy the file dl_field.control to your DL_FIELD installation folder, e.g.:

```
> cp dl_field.control ~/Desktop/Software/dl_field_4.3
```

Move to this directory, edit the file dl_field.control, read it and try to understand the main options. You may need to consult the manual (dlf_manual_4_3.pdf) to understand all the details, but for now, just give the right path to the ethanol.xyz file in Case1, and change also the path for the output PDB file that will be generated. Run DL_FIELD:

```
> ./dl_field
```

and check the output generated in the directory dl_field_4.3/output.

The second line of the file dl_field.control allows choosing the forcefield (FF), so in this case, we have generated a FIELD for one molecule of ethanol using the OPLS2005 potential. A CONFIG file containing also a single molecule in a cubic box of side L = 5 Å is also generated. But we need to generate a disordered configuration containing several ethanol molecules (let's say about 100). Edit the dl_field.control file, change the cell vector dimensions to 21 Å and in the Solution Maker line (line 11) give the following parameters:

```
1    100   molecules   1.8
```

This will create a CONFIG file containing the positions of 100 ethanol molecules placed in a cubic box of side L = 22 Å and having a minimum distance of 1.8 Å between them. Copy the two output files generated by DL_FIELD (output/dl_poly.FIELD, output/dl_poly.CONFIG) to your working directory (Case1) as FIELD and CONFIG, respectively. Edit them to see their content and try to understand it.

Load also the CONFIG file in Avogadro to visualize it[7].

So now we already have a simulation box with a reasonable (even if small) number of molecules at a not too bad density (≈0.72 g/cm$^3$, while the experimental density of liquid ethanol at 293 K is 0.789).

[7] Avogadro is able to open DL_POLY's CONFIG files. However, as you can see, in this case the graphic is not too nice, as the bonds are missing and the hydroxyl hydrogen is misinterpreted as an holmium atom due to the naming employed. You can convert the CONFIG file to the PDB format using Open Babel: `> obabel -i CONFIG CONFIG -o pdb -O CONFIG.pdb,` but then you still have to edit the pdb file and replace 'HO' by ' H'. Do this and reload the file in Avogadro and you will see a nicer plot.

The next step consists then in equilibrating our system at the desired thermodynamic conditions, so we will start by performing an NPT simulation. Edit the file control_equil_npt, try to understand what it does and save it with the name CONTROL in the same directory as the files CONFIG and FIELD. Go to this directory and run DL_POLY, e.g.:

```
> DLPOLY.X
```

This will run a short equilibration run of 10 ps that should be completed in 2-3 minutes, even if you are using a single processor.

The java interface included with DL_POLY can be used to read and plot the data contained in the output files generated by DL_POLY, so we can use it to follow the evolution of the simulation. Launch it as: `> dl-gui` and then select Analysis → Statistics and a property to plot. You can easily check how the temperature, the pressure, the volume or the different components of the energy evolve during the simulation and determine when you have reached equilibrium (You certainly will need to perform a much longer run to equilibrate correctly our system).

You can also use other alternative forcefields. For example, try pcff or cvff (change the keyword on the second line of the file dl_field.control) and compare the different resulting FIELD files.

The list of available FFs appears at the bottom of dl_field.control (or in the manual: ~/Desktop/Software/dl_field_4.3/dlf_manual_4_3.pdf), but only opls2005, pcff, and cvff can be used with input configurations using the XYZ format. In general, it is better to use PDB files as input, as they contain additional information. So let's try now to start from a PDB file. We already generated one when creating our previous configuration. We can visualize it with Avogadro (or Rasmol, VMD or any molecular visualizer, as the PDB format is a very popular standard format). Therefore this can be a representative example of a typical case in which someone has sent us a PDB file with a representative configuration of the system that we want to simulate or we have downloaded a PDB base from one of the available structural data bases[8].

Modify the dl_field.control file to read this PDB file and try to convert it. What happens? The conversion will fail because DL_FIELD cannot identify a molecule named MOL. This is the keyword appearing in the 4$^{th}$ column of the PDB file and corresponds to the name that identifies the molecule or residue. It has to match one of the names used in the DL_FIELD data bases in order that DL_FIELD can identify it correctly. Open the file ~/Desktop/Software/dl_field_4.3/lib/DLPOLY_OPLS2005.sf and search the word 'ethanol'. You will find that in the section MOLECULE_TYPE there is an ethanol molecule and that its associated key is ETOH. You can also see below that there is a molecular description of the molecule, indicating the atoms that compose it, their types and their connectivity. If the system that you want to simulate is not included in any of the available molecular structure (.sf) files, then you will need to add it before trying to use DL_FIELD. For example, now we

[8] For example, from the protein data bank: https://www.rcsb.org.

could create a MOL entry and simply copy the existing ethanol as our MOL molecule (we will see this later). But now it is more convenient to edit the file and replace the meaningless MOL keyword by the more meaningful name of ETOH. Do this and run DL_FIELD. You can also replace the XYZ name by your molecule name (e.g. Ethoh[9]). This is the name that will appear in the Molecule entry in the generated FIELD, but this can also be easily modified manually directly in the file.

Check now the output files. You will see that the CONFIG file looks very similar to the one that we obtained before, but the order of the atoms has changed and corresponds to the order given by the molecular definition used in the .sf file. Additionally, if you have saved also a new PDB file, you can load it in Avogadro and you will find that now the representation is correct (i.e., we don't have any more the problem of the HO atoms being mistaken as holmium). The generated FIELD will be the same as before. You can try again to check what happens when using other FFs. For example, now the CHARMM22_prot and CHARM36_cgenff FFs become available.

[9] In the PDB file you have a limit of 6 characters to choose the name and you have to pay attention to not modify the column numbers of the other entries (e.g. the element symbol has necessarily to appear in columns 77-78).

5. **Case 2: TIP4P/2005 water**

Here we are going to see how to create a new molecule definition in DL_FIELD, but first, we will prepare a system using one of the available water definitions already existing. In the provided OPLS2005 FF, there are several water potentials, but we will use the TIP4P one. Give a look to the dl_field_tip4p.control file and run DL_FIELD to generate the output files.

The procedure is similar to the one we employed with ethanol, but this is a rigid model (so we do not have any intramolecular degree of freedom). Additionally, it contains also a massless pseudo-site not located in any of the atoms and used to place the negative charge associated to the oxygen. This pseudo-site has to have the same name in the PDB file that in the description given in the .sf file (Q4 in our case). Additionally, we have to indicate in the dl_field.control that we want to use rigid bodies (line 25) and define the rigid bodies below, with the command RIGID W1, where W1 refers to the molecule or residue name given in columns 69-75 of the PDB file[10].

You can check the output files generated by DL_FIELD and run a short simulation to equilibrate the system and test the use of rigid bodies.

However, we would like to simulate water using an improved version of the TIP4P potential, which is the TIP4P/2005 model proposed by Abascal and Vega (J. *Chem. Phys.* **123**, 234505 (2005)). It is very similar to the original TIP4P, but the O-Q4 distance is slightly larger and the partial charges and Lennard-Jones parameters are also slightly different.

We can start by editing the h2o.pdb file in order to change the distance between the O atom and the pseudo-site, which now is 0.1546. We also have to change the residue name in columns 18-21, as TIP4 corresponds to the original model. Let's call our water TIP4P/2005 as WT05, so the new PDB file should look like this:

```
ATOM      1  O   WT05    1       0.000   0.000   0.000           W1
ATOM      2  H   WT05    1       0.757   0.586   0.000           W1
ATOM      3  H   WT05    1      -0.757   0.586   0.000           W1
ATOM      4  Q4  WT05    1       0.000   0.155   0.000           W1
```

If you try to run DL_FIELD now, the program will complain that it does not recognize the keyword WT05, so first, we need to create a user-defined forcefield (udff) file and give the required information there. You can try to do this from scratch by editing a new file and using the TIP4P model described in the file dl_field_4.3/lib/DLPOLY_OPLS2005.sf. Otherwise, give a look to the water.udff file already present in the Case2 directory. You can see that we are adding a new entry for the OPLS2005 potential, that we define two new types of atoms (O_tip4p2005 and HO_tip4p2005) and a new molecule type (tip4p2005_water). We cannot

[10] In case DL_FIELD complains about 'Fail to determine corresponding OPLS2005 index for ATOM_TYPE Q4_pseudo of index 4' you will need to edit the file ~/Desktop/Software/dl_field_4.3/lib/dl_field.atom_type and add it under the section DL_FIELD OPLS2005 as Q4  Q4  Q4_pseudo.

repeat any atom name, but we may have repeated atom keys (e.g. OW and HW in our example) if we can use the same VDW parameters. The molecule type and key must be unique. Then we define the molecule and its connectivity with the entry MOLECULE and give the partial charges corresponding to each site (here +0.5564 for each H atom and -1.1128 in the lone pair site of the oxygen). Finally, in the VDW entry, we give the Lennard-Jones parameters of our two new atoms.

The dl_field.control has to be modified to tell DL_FIELD to charge this new molecule. This is simply done by giving in line 7 the path to the udff that we just created. Finally, you will also need to add the two new atoms to the library file dl_field_4.3/lib/dl_field.atom_type. Edit this file, search for OPLS2005 and add this two lines to the existing atom list:

```
OW      O133    O_tip4p2005
HW      H134    HO_tip4p2005
```

You are ready now to create the CONFIG and FIELD input files using DL_FIELD and then do your own simulation of this popular water model.

6. **Case 3: Benzene crystal**

We are going to see now how to prepare a different type of system, not a liquid, but a molecular crystal. In this case, the most likely input is a Crystallographic Information File (CIF). Most molecular visualizers read CIFs and many allow you to manipulate them, generate a supercell and export the atomic positions into other formats. For example, you can use Avogadro to open the file benzene_2100350.cif, which corresponds to the structure of phase I of crystalline benzene measured at 296 K and 0.3 GPa (A. Budzianowski and A. Katrusiak, *Acta Cryst. B* **62**, 94-101 (2006)).

However, using Avogadro there is no straightforward way of generating a PDB file that maintains the order of the atoms keeping them grouped by molecules. Therefore we will use an alternative program, Mercury, developed by the Cambridge Crystallographic Data Centre (CCDC) specifically to handle crystal structures. Launch Mercury (you will find it in the Software folder in your Desktop) and open the CIF for benzene. You will see a single benzene molecule, but selecting the Packing option in the Display window you will see the unit cell of the crystal. We need to generate a larger supercell for our simulation, so go to the menu Calculate and select "Packing/Slicing ...". You can use directly the option 3x3x3 and you will generate a supercell formed by 27 unit cells, that you can then save as a PDB file, as we need a PDB to work with DL_FIELD. However, if you edit the file you will see that there are too many atoms. The unit cell contains Z=4, so we will expect to have 108 benzene molecules in our supercell, i.e. 1296 atoms, and instead, we have 2064. This is because Mercury generates also the image molecules that are in the corner or the sides of the supercell. You can see this more easily by looking to the unit cell and verifying that it contains much more than 4 molecules. You can solve this easily by going back to the Packing and Slicing interface and using 2.9 instead of 3 for the packing along a, b, and c. Save the new supercell as benzene_supercell.pdb and check that now it contains 1296 atoms. Edit the file and change also the residue name from UNK to BENZ.

Now we are ready to use DL_FIELD to generate our input files for DL_POLY, using the same procedure as before. In particular, note that now it's very important to give the right dimensions of the simulation box as 3a, 3b, and 3c, where a= 7.243 Å, b=9.31 Å, and c=7.656 Å are the unit-cell dimensions.

Check the files generated by DL_FIELD and run a short simulation with DL_POLY to check if everything works as it should.

The FIELD file generated is correct and contains the needed improper terms to maintain benzene planar (see benzene's definition in DLPOLY_OPLS2005.sf). In this case, they will appear in the FIELD file as a set of dihedral terms with very large force constants. If we are only interested in the global dynamics of the molecule and we don't want to bother with fast intramolecular motions, an alternative is to define the whole molecule as a single rigid body. Try it!

Hints: You will need to redefine the benzene molecule using a UDFF. You also need to give a molecule name in the PDB file (columns 69-75) and use this name in the rigid directive in dl_field.control.

You can then run the simulations using the rigid molecules and compare the results with those obtained before using the flexible model.