Exercise 2 part 2 - Parameterising a new molecule (valence & vdW)

Contents

Remaining parameter setup and free energy calculations

Assigning valence parameters in AMOEBA (bonds, angles, stretch-bends, out-of-plane bends, torsions & vdW parameters) is similar to the process used in other force fields. There are a few potential options:

- Assign by similarity to existing parameters
- Derive new parameters directly from QM data
- A mixture of the two (e.g. by refining an initial set of parameters with a QM torsional energy profile)

Valence parameters in AMOEBA are defined by atom class (see side note 1 for the difference between atom class and atom type), and a variety of atom classes already exist for organic molecules (see `amoeba09.prm` in the Tinker `params` folder) or proteins (see `amoebapro13.prm`). If you wish to use these, they can be sourced in your simulations *via* the use of the parameters keyword at the top of your key file. If you've defined new atom classes, or if you wish to refine these parameters further, the Tinker `valence` program is the way to go.

Side note 1 - Atom types vs atom classes

If you remember the layout of our AMOEBA prm file from our electrostatics parameterisation, there were a few lines beginning 'atom' at the top of the file, e.g.:

```
atom        401  401    C      "ethanol"
```

The second and third column, here containing `401`, define the atom type and atom class respectively. Atom types are used for electrostatics parameters - each unique set of multipoles/polarisabilities should have a unique atom type. Atom classes are used for valence & vdW parameters - multiple different atom types may have the same atom class and hence the same bonds/angles/dihedral parameters or vdW sigma/epsilon values.

The `valence` program

Assigning some initial parameters

Let's continue in the `Exercise_2` folder in which we've created our AMOEBA prm file with a final set of multipoles and other electrostatic parameters. However, to keep things tidy we'll start with fresh filenames:

```
cp ethanol_renumbered.xyz ethanol_valence.xyz
cp AMOEBA_ethanol_postfit.prm AMOEBA_ethanol_valence.prm
```

Next, run the `valence` program interactively (just type `valence` on the command line) and you'll see four options:

```
  The TINKER Valence Parameter Facility Can :

     (1) Set Initial Values for Valence Parameters
```

翻譯

```
      (2) Compare QM and MM Vibrational Frequencies
      (3) Force Fit of Parameters to QM Results
      (4) Structure Fit of Parameters to QM Results
```

Today we'll only be using option 1, as ethanol is a fairly simple molecule, but as you can see valence can also be used to refine bond/angle parameters more tightly to recreate QM data. Option 3 minimises AMOEBA forces for a given QM structure, and option 4 minimises the structural difference between QM & MM minimum structures.

Choose option 1 and follow the instructions on screen, using the ethanol_valence.xyz set of coordinates, the ethanol_esp.log Gaussian output file and the AMOEBA_ethanol_valence.prm set of input parameters. Valence prints out a set of estimated vdW and bonded parameters to screen for you. Frustratingly they're not printed to file! You may wish to re-run the program non-interactively and redirect output to a new file, e.g.:

```
valence 1 ethanol_valence.xyz ethanol_esp.log AMOEBA_ethanol_valence.prm >
valence_estimates.txt
```

Add the vdW, bond, angle, stretch-bend and torsional parameters to the AMOEBA_ethanol_valence.prm file and your parameterisation of ethanol is complete. For more complicated small molecules than ethanol one should check that the torsional parameters accurately represent molecular energetics as a bond rotates - the Tinker torsfit program can be used to fit torsions to QM data, for example. For now, though, we'll use the suggested torsional parameters for ethanol and perform some basic simulations with our molecule to validate the parameters.

Parameter validation - minimisation

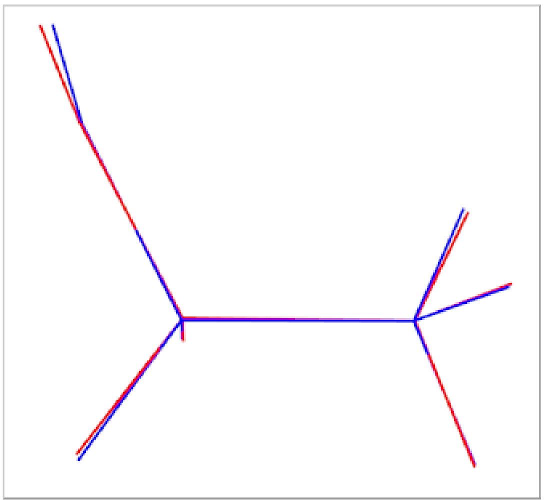
Comparing the MM and QM minimised structures is a simple way to ensure our newly assigned parameters don't distort the molecular geometry or move too far from the local QM minimum. Create a new key file containing options to run a minimisation similarly to exercise 1, e.g.:

```
parameters AMOEBA_ethanol_valence.prm

openmp-threads 1

cutoff 999.0
polar-eps 0.00001

neighbor-list

maxiter 2000
printout 100
```

Run this minimisation on the login node - it's extremely quick so we don't need to submit a job to Lyceum:

```
minimize ethanol_valence.xyz -k ethanol_min.key 0.01
```

You can copy the resulting file, ethanol_valence.xyz_2, back to your local desktop using WinSCP and open it in VMD. If you copy the initial structure, ethanol_valence.xyz, back too and compare it with the final structure they should look something like this:



Where the initial structure is in blue, and the final in red. As you can see, there is little difference between the MM and QM minima, which hopefully suggests a high degree of confidence in using these parameters for simulations. We will test this in the next step.

Setting up a solvation free energy calculation

Solvating our system

To run a solvation free energy calculation we'll first need to solvate and equilibrate our system in a box of water molecules. AMOEBA uses a 3-site fully flexible water model, so in theory we could use any pre-equilibrated box of 3-site waters for this, but Tinker comes packaged with water boxes of a variety of sizes, so we will use one of those instead (you'll find it in your folder, called `waterbox.xyz`. As the header describes, this is a cubic box of side length 24.662 A). We'll use the program `xyzedit` to facilitate this. Run `xyzedit` with the minimised ethanol structure and key file:

```
xyzedit ethanol_minimised.xyz -k ethanol_min.key
```

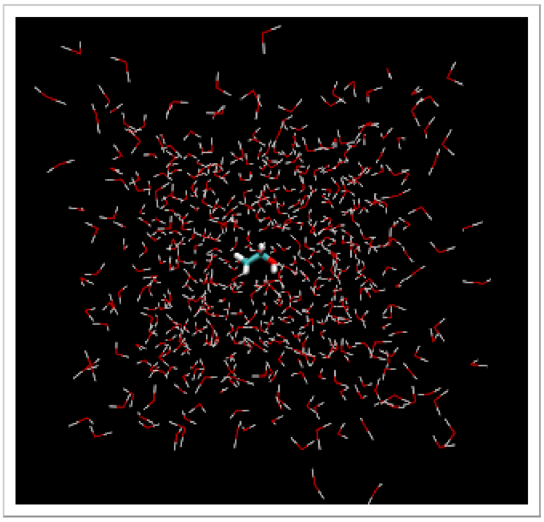
You'll see a huge range of options:

```
 The TINKER XYZ Editing Facility can Provide :

    (1) Offset the Numbers of the Current Atoms
    (2) Deletion of Individual Specified Atoms
    (3) Deletion of Specified Types of Atoms
    (4) Deletion of Atoms outside Cutoff Range
    (5) Insertion of Individual Specified Atoms
    (6) Replace Old Atom Type with a New Type
    (7) Assign Connectivities for Linear Chain
    (8) Assign Connectivities based on Distance
    (9) Convert Units from Bohrs to Angstroms
   (10) Invert thru Origin to give Mirror Image
   (11) Translate All Atoms by an X,Y,Z-Vector
   (12) Translate Center of Mass to the Origin
   (13) Translate a Specified Atom to the Origin
   (14) Translate and Rotate to Inertial Frame
   (15) Move to Specified Rigid Body Coordinates
   (16) Move Stray Molecules into Periodic Box
   (17) Delete Molecules outside of Periodic Box
   (18) Append a Second XYZ File to Current One
   (19) Create and Fill a Periodic Boundary Box
   (20) Soak Current Molecule in Box of Solvent
```

First choose option 12, then choose option 20, which will centre and solvate your system, automatically deleting any water molecules that overlap with the solute. When prompted after selecting option 20, enter the name of the solvent box (`waterbox.xyz`) and take note of the new filename that has been written out - rename this to something more sensible, e.g.:

```
mv ethanol_minimised.xyz_2 ethanol_solv.xyz
```

If you wish, you can copy the solvated system back to your local desktop and visualise it in VMD to check it has been solvated correctly - you should see something like this:



Once happy the solvated system looks sensible you can move on to the next stage, in which we'll minimise and equilibrate the system as a whole, ready for input to a solvation free energy calculation.

Minimising and equilibrating

Minimising and equilibrating the system will be performed using a similar protocol to that we used in Exercise 1. First, however, we'll have to combine our ethanol parameters with some water parameters in our prm file:

```
cp AMOEBA_ethanol_valence.prm AMOEBA_ethanol_solv.prm && cat outputs/water03.prm
>> AMOEBA_ethanol_solv.prm
```

The `water03.prm` file in the outputs folder is an edited version of that available in the Tinker `params` folder, with the usual AMOEBA header of force field options deleted as we already have these in our `AMOEBA_ethanol_valence.prm` file.

You may use the key files from Exercise 1 as a guide for preparing your own (remember you'll need to update the periodic box dimensions to the correct size for our system here - a cubic box of sides 24.662 A, and source our ethanol+water parameters at the top of the file using the `parameters` keyword), or you can find suitable inputs available as `ethanol_solv_min.key`, `ethanol_solv_nvt.key` and `ethanol_solv_npt.key` in the `outputs` folder.

A script to submit the minimisation and equilibration to Lyceum is available, which will run each MD simulation for a total of 10 ps. Submit this with:

```
qsub Equilibrations.x
```

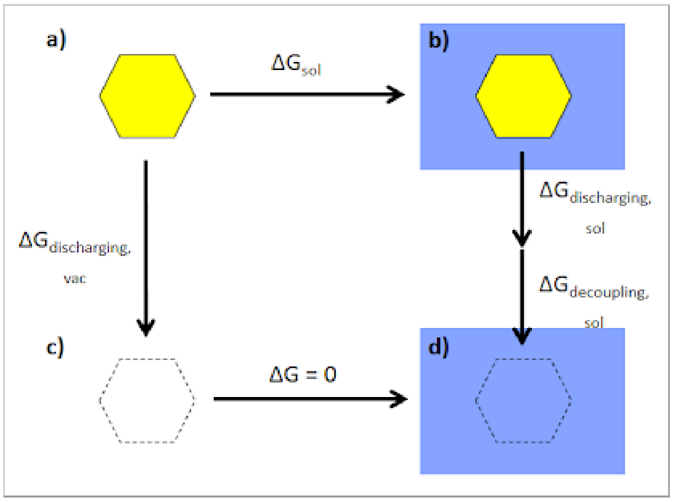
This should take around 15-20 minutes to run, so you may wish to grab a coffee during this time. Alternatively, if time is tight, feel free to reduce the length of the simulations by modifying `Equilibrations.x`.

Once the simulations are complete, you can look at the output files to check temperature & pressure have equilibrated (these simulations are quite short, so don't worry if they're not completely equilibrated yet), and/or copy the trajectory files back to your local machine to visualise them in VMD.

As a final step in this exercise, we'll use our final equilibrated system as the starting point for our solvation free energy calculations.

Solvation free energy setup

As a reminder, we'll be recreating the thermodynamic cycle below:


a)
ΔG_{sol}
b)
$\Delta G_{discharging,\ sol}$
$\Delta G_{discharging,\ vac}$
$\Delta G_{decoupling,\ sol}$
c)
$\Delta G = 0$
d)

As such, we'll have to set up simulations that describe the transformations **a -> c** and **b -> d**, i.e. a removal of charges on the solute in both solution phase and gas phase, and a vdW decoupling of the solute from its environment in the solution phase only (in the gas phase there is no environment, so the free energy of decoupling is necessarily zero!). For choice of number of intermediates/lambda window spacing we will follow a variety of published studies and use linear spacing for the discharging step and non-linear spacing for the vdW decoupling:

- Lambda windows for electrostatics = 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.0
- Lambda windows for vdW = 1.0, 0.9, 0.8, 0.75, 0.7, 0.65, 0.6, 0.5, 0.4, 0.2, 0.0

Tinker has a simple functionality for absolute free energy calculations, including softcore vdW treatment of appearing or disappearing atoms, controlled by specifying a few keywords in the key file:

```
ligand -1 9              # Definition of atoms to be mutated (negative first
number defines a range)
ele-lambda 0.9           # Lambda value for scaling electrostatic interactions
(1.0 = on, 0.0 = off)
vdw-lambda 1.0           # Lambda value for scaling vdW interactions (1.0 = on,
0.0 = off)
```

Adding the above lines to a key file scales the multipoles and polarisabilities on atoms 1-9 (inclusive) by 0.9, but keeps the vdW interactions completely unperturbed. We therefore need to create a separate key file for each different lambda value, in solution phase and gas phase. To do this by hand would be a laborious process, so we'll use a basic script to do this for us:

```
./create_ele_vdw.x
```

Provided you've used the default filenames we've suggested for everything throughout (if not, try to make some edits to the script), this should create two folders, `solution` and `gas`, containing subfolders for each lambda window and everything needed to run the simulations. Take a look around and notice how the `ele-lambda` and `vdw-lambda` keywords change for each simulation. Also note we're using a far tighter convergence on our induced dipoles (`polar-eps 0.00001`) as tightly converged dipoles will improve the accuracy & precision of free energy differences between states.

These simulations are all ready to run. However, each lambda window (in solution phase) will take somewhere around 12-14 hours, so we won't start them today. Instead, in the next exercise we'll use an example set of trajectories to perform a BAR free energy analysis.

註解