AMOEBA advanced potential energies workshop

Search this site

Navigation

- Home
- Intro & setup
- Exercise 1 - Getting started with Tinker
- **Exercise 2 part 1 - Parameterising a new molecule (electrostatics)**
- Exercise 2 part 2 - Parameterising a new molecule (valence & vdW)
- Exercise 3 - Analysing trajectories and calculating free energies
- Sitemap

Exercise 2 part 1 - Parameterising a new molecule (electrostatics)

Contents

Available AMOEBA parameters

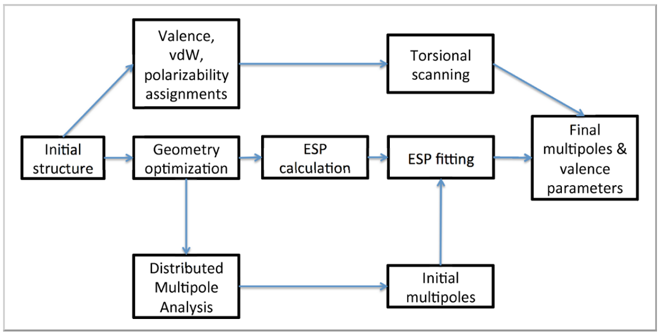
AMOEBA parameters for biomolecular simulations cover proteins, nucleic acids, water, ions and some small molecule organic solutes. However, for the vast majority of protein-ligand simulations, or if you wish to study further biomolecular moieties, you will need to add new parameters that are consistent with the rest of the force field. While this is conceptually very similar to parameterisation processes for any other force field, the intricacies are slightly different, and will be explored here.

In this exercise we'll use ethanol as an example molecule to parameterise, and set up a series of simulations as if we wished to perform a hydration free energy calculation. This will take a number of steps, so this exercise will be split over lunch.

Parameterising ethanol

AMOEBA parameterisation schematic

The overall AMOEBA parameterisation process generally follows the scheme below, a more detailed description of which can be found in the supplementary information of "Polarizable Atomic Multipole-Based Molecular Mechanics for Organic Molecules" by Ren, Wu & Ponder, 2011:


Valence, vdW, polarizability assignments
Torsional scanning
Initial structure
Geometry optimization
ESP calculation
ESP fitting
Final multipoles & valence parameters
Distributed Multipole Analysis
Initial multipoles

Starting from just an initial structure (on the left-hand side of the above) we can assign all required bonded and non-bonded parameters using a variety of Tinker utilities. The overall workflow we'll use is as follows (you can compare this with the flowchart above):

1. QM **geometry optimise** the initial structure at the MP2/6-311G** level of theory
2. Perform a **Distributed Multipole Analysis** to generate an initial set of atomic multipoles
3. Refine atomic multipoles by **fitting to molecular electrostatic potential** at the MP2/aug-cc-pVTZ level

翻譯

4. Assign initial **bond, angle, dihedral and vdW parameters** using the Tinker `valence` program
5. [Optional] **Refine valence parameters** to QM-derived intramolecular forces, geometries or torsional scanning profiles
6. **Run a simulation** to validate parameters, re-refine if necessary

We'll start with the most important parameters in AMOEBA - the permanent atomic multipoles.

Multipole parameter assignment 1: Geometry optimisation

Just as for common fixed-point charge force fields, AMOEBA derives its classical electrostatics parameters from QM calculations. However, the levels of theory and basis sets used in these calculations are generally higher and larger than those of most fixed-charge force fields, as large basis sets including polarisation and/or diffuse functions are necessary to accurately capture electrostatic potentials around many molecules. It it these electrostatic potentials that AMOEBA must accurately recreate with a multipole + induced dipole model.

We'll start with step 1, an MP2 geometry optimisation using the 6-311G** basis set. Throughout, we'll use Gaussian 09 as our QM package because some Tinker utilities are set up to parse Gaussian output files, however you may use your own software if you prefer. First, log into Lyceum and change into the directory `Exercise_2`. Inside you'll find two Gaussian input files called `ethanol_hf_opt.com` and `ethanol_dma_opt.com`. These will perform the QM geometry optimisation, but split into two parts - first optimising at a low level of theory (HF/6-31G*), and then re-optimising the final structure at our desired, higher, level of theory (MP2/6-311G**). This is often faster than simply immediately performing the optimisation at the higher level of theory.

Take a look at the contents of the fist file, `ethanol_hf_opt.com`. If you're familiar with Gaussian inputs the contents should be fairly obvious, but if not, the line of interest is:

```
#opt=(maxcycle=400) HF/6-31G* MaxDisk=10GB
```

This requests a geometry optimisation ('`opt`'), with a maximum of 400 optimisation steps ('`maxcycle=400`'), using Hartree-Fock theory and the 6-31G* basis set ('`HF/6-31G*`'), and a maximum of 10 GB of hard disk space usage for temporary scratch files ('`MaxDisk=10GB`'). The rest of the file contains details of the amount of memory and processors to use for the job, charge and multiplicity information, and Cartesian coordinates of the initial structure. This structure can be created, for example, interactively from a SMILES string and the babel software (but fortunately we've already done this for you today!):

```
echo CCO > ethanol.smi
babel -ismi ethanol.smi -ogau ethanol_gaussian.com --gen3d
```

In the second Gaussian input file, `ethanol_dma_opt.com`, we perform the second optimisation at a higher level of theory:

```
#opt=(maxcycle=400) MP2/6-311G(1d,1p) Density=MP2 MaxDisk=10GB Geom=AllCheck
```

This time we use MP2 theory for both performing the simulation and any analysis/results in the output ('`Density=MP2`'), the 6-311G** basis set ('`MP2/6-311G(1d,1p)`'), and read in the geometry, charge & multiplicity information from the previous checkpoint file ('`Geom=AllCheck`').

A script has been set up to run both optimisations, called Optimisations.x. Submit it to Lyceum:

```
qsub Optimisations.x
```

Fortunately, since ethanol is so small, this should only take a minute or so to run. Larger molecules will require much more time.

Multipole parameter assignment 2: Distributed Multipole Analysis

AMOEBA uses Stone's original DMA procedure to generate the initial set of permanent atomic multipoles at atomic sites (for a brief explanation of DMA see side note 1). To carry this out we'll make use of the GDMA program and a variety of Tinker programs.

To start with, we need to get our QM electron density calculated by Gaussian into an amenable format for manipulation by other programs. Convert the binary Gaussian checkpoint file into an ASCII human-readable file with `formchk`:

```
formchk ethanol_dma_opt.chk
```

Side note 1 - What is DMA?

A Distributed Multipole Analysis (or DMA) is a procedure to describe the charge distribution in a molecule via a multipole expansion about a series of defined centres. In AMOEBA it's used to derive an initial set of permanent atomic multipoles for a molecule. However, because

The output formatted checkpoint file (called `ethanol_dma_opt.fchk` by default) will be one of the inputs to GDMA. The other is a text file with details of the calculation options. Open a new file `ethanol.gdmain` and type the following inside, **without the comments**:

DMA multipoles can be highly conformation and basis set dependent, this initial set of multipoles is later refitted to recreate a further QM calculation of electrostatic potential outside the molecule. For more details on how DMA really works, Stone's "Theory of Intermolecular Forces" is an excellent starting point.

```
Title "Ethanol at MP2/6-311G(1d,1p)"

Verbose
Density MP2                # Use MP2 electron
density
File ethanol_dma_opt.fchk  # File containing
electron density

Angstrom                   # Units specification
AU

Multipoles
Switch 0                   # Use Stone's original
(1.0) DMA method
Radius H 0.65              # Set H radius ratio
equivalent to other atoms
Limit 2                    # Limit multipole
order to quadrupoles
Start

Finish
```

This sets up GDMA to perform Stone's original DMA procedure on the MP2 electron density from the `ethanol_dma_opt.fchk` file. Next, we just need to run GDMA, making sure to redirect the output to a new file:

```
gdma < ethanol.gdmain > ethanol.gdmaout
```

The `ethanol.gdmaout` file now contains the monopole, dipole and quadrupole components of the multipole at each atomic site, oriented in the global x,y,z coordinate frame. In the next step we'll convert these into Tinker format parameters and define multipole local frames, polarisabilities and polarisation groups. So what are local frames, polarisabilities and polarisation groups...?

Multipole parameter assignment 3: Local frames, polarisabilities and polarisation groups

1. **Local frames:** While we now have multipole components oriented in a global frame, this global frame would be different if we rotated our molecule, used a different starting conformation or even solvated our molecule in an existing, independently oriented, box of water. For practical application therefore, it makes more sense to define a local frame for each of our multipoles that can be rotated as the molecule tumbles and changes conformation during simulation (for more information on how local frames can be chosen, see side note 2).

2. **Atomic polarisabilities:** Within AMOEBA the induced dipoles calculated at every atomic site are a function of the electric field at that site **E**, and the atomic polarisability alpha:

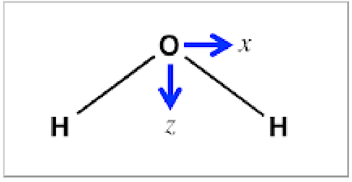
$$\vec{u} = a\vec{E}$$

A polarisability therefore needs to be defined for every single atom in the system. In practice there are a number of suggested AMOEBA atomic polarisabilities that cover a range of chemistries, and that originate with Thole. There are exceptions for aromatic carbon and hydrogen atoms, which have slightly larger polarisabilities than their aliphatic counterparts.

3. **Polarisation groups:** A 'polarisation group' in AMOEBA terms defines a contiguous group of atoms whose permanent multipoles do not polarise one

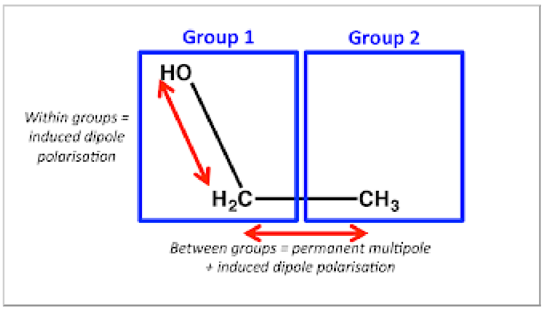
Side note 2 - Local multipole frames

Local frames are necessary to correctly reorient and rotate atomic multipoles as molecules tumble. Bonds between atoms can provide convenient local axes for defining these frames, but choice of local frame is important for correctly representing electrostatic interactions upon conformational change. Atoms within molecules may be related by symmetry - water, for example, has C_{2v} symmetry. Motion of both O-H bonds should therefore affect the rotation of the central oxygen multipole. This is achieved by defining the z-axis along the bisector of the H-O-H angle, and the x-axis in the plane of the molecule:



Without this bisector axis definition, conformational change in one O-H bond would affect the orientation of the central multipole more than the other. For a full description of local frame definitions in AMOEBA see relevant development papers.

another. Between polarisation groups, permanent multipoles **and** other induced dipoles contribute to the field experienced by an atom, and hence polarise the induced dipole at that atom. Within a polarisation group, the other permanent multipoles are **excluded** from contributing to the field at an atomic site, and hence **only** the other induced dipoles polarise the atom in question. Polarisation groups are roughly chosen to cover a single functional group that is typically conformationally rigid. The rigidity means that the intra-group polarisation will be fairly constant, and can be well-described by the permanent multipoles on each of the atoms. For ethanol, we'll choose polarisation groups as follows:


Group 1
Group 2
HO
H_2C
CH_3
Within groups = induced dipole polarisation
Between groups = permanent multipole + induced dipole polarisation

The Tinker `poledit` program is designed to create a set of AMOEBA multipoles from a GDMA output. We'll run it interactively here. Simply type

```
poledit
```

at the command prompt and follow the instructions on screen. This will take you through the following steps:

1. Reading in the GDMA output (choose `poledit` option 1 and enter the name of your GDMA output file (`ethanol.gdmaout`) when requested)
2. Choosing local frames (defaults are suggested...they should be ok for ethanol, but check you understand which atoms are defining the axes)
3. Choosing atomic polarisabillity values (again defaults are suggested - the full list of AMOEBA defaults can be found in the supplementary information of this paper)
4. Choosing polarisation groups (hint - for ethanol we should define two polarisation groups as above, separated along the C-C bond)
5. Averaging multipoles of equivalent atoms, e.g. methyl hydrogens (generally recommended, but **not** performed by default, you'll have to override the default '`N`' option)
6. Removing multipoles that should be zero due to symmetry, e.g. *x* & *y* dipole components of water oxygen (generally recommended, but again **not** performed by default)

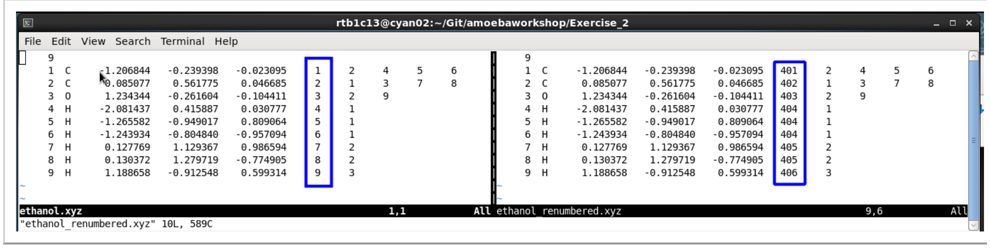
In addition to the details you can see on screen, `poledit` will create a Tinker xyz file and key file containing coordinates and the multipole & polarisability parameters you've defined - if you've followed our naming scheme above these will be called `ethanol.xyz` and `ethanol.key`.

Multipole parameter assignment 4: Tidying up atom types

By default, poledit assigns unique atom types for every single atom in the input structure, numbered sequentially starting with atom type 1. This numbering assumes firstly that we won't be mixing our parameters with other parts of the AMOEBA force field (which may also have atom types numbered 1 -> N), and secondly that there are no atoms we wish to have identical atom types (such as methyl hydrogens).

Both of these are untrue for the hydration free energy calculation we're setting up here - we want to mix our ligand parameters with those of water, AND we want some of our hydrogens to be chemically equivalent. Therefore we need to renumber the atom types in our xyz and key files, and remove any duplicate sets of parameters.

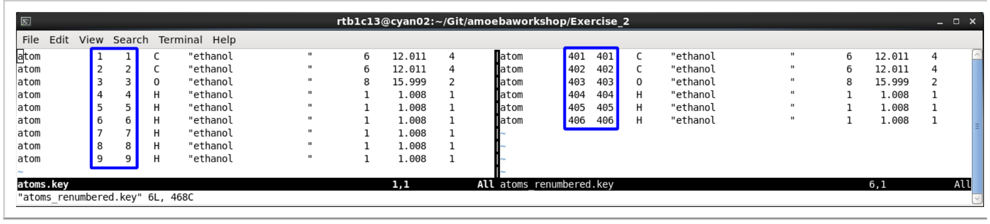
The safest way to do this, avoiding any chance of clashes with existing atom types in other AMOEBA parameter files, is to simply add a constant to our ethanol atom type numbers. The largest atom type in the 2013 AMOEBA protein force field (`amoebapro13.prm`) is 258. Therefore we'll renumber starting at 401 to avoid any clashes. We'll also assign identical atom types to the CH_3 hydrogens and independently to the CH_2 hydrogens. Remember in the xyz file the atom types are in the 6th column. Below you can see example xyz files before (left) and after (right) renumbering - there's also a copy in the `outputs` folder if you lose track of the changes you make:


rtb1c13@cyan02:~/Git/amoebaworkshop/Exercise_2
File Edit View Search Terminal Help

```
     9
     1  C     -1.206844   -0.239398   -0.023095     1     2     4     5     6
     2  C      0.085077    0.561775    0.046685     2     1     3     7     8
     3  O      1.234344   -0.261604   -0.104411     3     2     9
     4  H     -2.081437    0.415887    0.030777     4     1
     5  H     -1.265582   -0.949017    0.809064     5     1
     6  H     -1.243934   -0.804840   -0.957094     6     1
     7  H      0.127769    1.129367    0.986594     7     2
     8  H      0.130372    1.279719   -0.774905     8     2
     9  H      1.188658   -0.912548    0.599314     9     3
~
~
ethanol.xyz                                        1,1           All
```

```
     9
     1  C     -1.206844   -0.239398   -0.023095   401     2     4     5     6
     2  C      0.085077    0.561775    0.046685   402     1     3     7     8
     3  O      1.234344   -0.261604   -0.104411   403     2     9
     4  H     -2.081437    0.415887    0.030777   404     1
     5  H     -1.265582   -0.949017    0.809064   404     1
     6  H     -1.243934   -0.804840   -0.957094   404     1
     7  H      0.127769    1.129367    0.986594   405     2
     8  H      0.130372    1.279719   -0.774905   405     2
     9  H      1.188658   -0.912548    0.599314   406     3
~
~
ethanol_renumbered.xyz                             9,6           All
"ethanol_renumbered.xyz" 10L, 589C
```

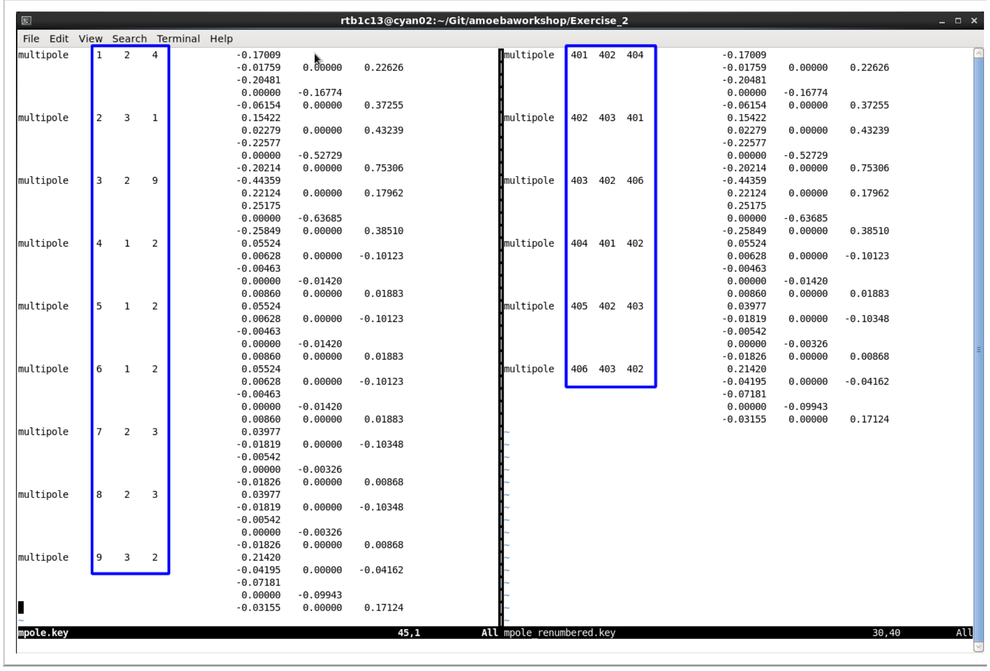
The key file also needs to be edited in the same way. There are three sections in the key file, defining atom types, multipole parameters and polarisability parameters. They should be renumbered as follows (again, there is a copy in the `outputs` folder if you lose track). First atom types:


rtb1c13@cyan02:~/Git/amoebaworkshop/Exercise_2
File Edit View Search Terminal Help

```
atom          1    1    C      "ethanol               "         6     12.011     4
atom          2    2    C      "ethanol               "         6     12.011     4
atom          3    3    O      "ethanol               "         8     15.999     2
atom          4    4    H      "ethanol               "         1      1.008     1
atom          5    5    H      "ethanol               "         1      1.008     1
atom          6    6    H      "ethanol               "         1      1.008     1
atom          7    7    H      "ethanol               "         1      1.008     1
atom          8    8    H      "ethanol               "         1      1.008     1
atom          9    9    H      "ethanol               "         1      1.008     1
~
atoms.key                                          1,1           All
```

```
atom        401  401    C      "ethanol               "         6     12.011     4
atom        402  402    C      "ethanol               "         6     12.011     4
atom        403  403    O      "ethanol               "         8     15.999     2
atom        404  404    H      "ethanol               "         1      1.008     1
atom        405  405    H      "ethanol               "         1      1.008     1
atom        406  406    H      "ethanol               "         1      1.008     1
~
~
~
~
atoms_renumbered.key                               6,1           All
"atoms_renumbered.key" 6L, 468C
```

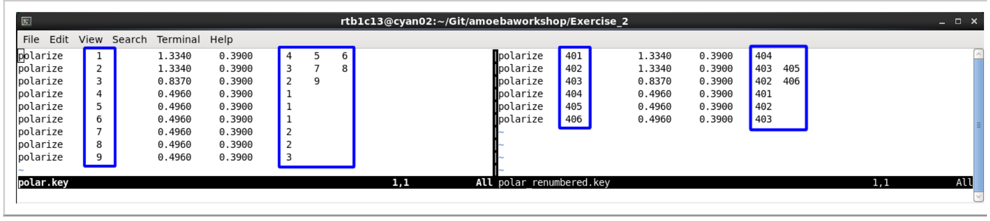
Multipoles:


rtb1c13@cyan02:~/Git/amoebaworkshop/Exercise_2
File Edit View Search Terminal Help

```
multipole     1    2    4                    -0.17009
                                             -0.01759    0.00000    0.22626
                                             -0.20481
                                              0.00000   -0.16774
                                             -0.06154    0.00000    0.37255
multipole     2    3    1                     0.15422
                                              0.02279    0.00000    0.43239
                                             -0.22577
                                              0.00000   -0.52729
                                             -0.20214    0.00000    0.75306
multipole     3    2    9                    -0.44359
                                              0.22124    0.00000    0.17962
                                              0.25175
                                              0.00000   -0.63685
                                             -0.25849    0.00000    0.38510
multipole     4    1    2                     0.05524
                                              0.00628    0.00000   -0.10123
                                             -0.00463
                                              0.00000   -0.01420
                                              0.00860    0.00000    0.01883
multipole     5    1    2                     0.05524
                                              0.00628    0.00000   -0.10123
                                             -0.00463
                                              0.00000   -0.01420
                                              0.00860    0.00000    0.01883
multipole     6    1    2                     0.05524
                                              0.00628    0.00000   -0.10123
                                             -0.00463
                                              0.00000   -0.01420
                                              0.00860    0.00000    0.01883
multipole     7    2    3                     0.03977
                                             -0.01819    0.00000   -0.10348
                                             -0.00542
                                              0.00000   -0.00326
                                             -0.01826    0.00000    0.00868
multipole     8    2    3                     0.03977
                                             -0.01819    0.00000   -0.10348
                                             -0.00542
                                              0.00000   -0.00326
                                             -0.01826    0.00000    0.00868
multipole     9    3    2                     0.21420
                                             -0.04195    0.00000   -0.04162
                                             -0.07181
                                              0.00000   -0.09943
                                             -0.03155    0.00000    0.17124
~
mpole.key                                          45,1          All
```

```
multipole   401  402  404                    -0.17009
                                             -0.01759    0.00000    0.22626
                                             -0.20481
                                              0.00000   -0.16774
                                             -0.06154    0.00000    0.37255
multipole   402  403  401                     0.15422
                                              0.02279    0.00000    0.43239
                                             -0.22577
                                              0.00000   -0.52729
                                             -0.20214    0.00000    0.75306
multipole   403  402  406                    -0.44359
                                              0.22124    0.00000    0.17962
                                              0.25175
                                              0.00000   -0.63685
                                             -0.25849    0.00000    0.38510
multipole   404  401  402                     0.05524
                                              0.00628    0.00000   -0.10123
                                             -0.00463
                                              0.00000   -0.01420
                                              0.00860    0.00000    0.01883
multipole   405  402  403                     0.03977
                                             -0.01819    0.00000   -0.10348
                                             -0.00542
                                              0.00000   -0.00326
                                             -0.01826    0.00000    0.00868
multipole   406  403  402                     0.21420
                                             -0.04195    0.00000   -0.04162
                                             -0.07181
                                              0.00000   -0.09943
                                             -0.03155    0.00000    0.17124
~
mpole_renumbered.key                               30,40         All
```

Polarisability parameters:


rtb1c13@cyan02:~/Git/amoebaworkshop/Exercise_2
File Edit View Search Terminal Help

```
polarize      1          1.3340     0.3900     4    5    6
polarize      2          1.3340     0.3900     3    7    8
polarize      3          0.8370     0.3900     2    9
polarize      4          0.4960     0.3900     1
polarize      5          0.4960     0.3900     1
polarize      6          0.4960     0.3900     1
polarize      7          0.4960     0.3900     2
polarize      8          0.4960     0.3900     2
polarize      9          0.4960     0.3900     3
~
polar.key                                          1,1           All
```

```
polarize    401          1.3340     0.3900     404
polarize    402          1.3340     0.3900     403  405
polarize    403          0.8370     0.3900     402  406
polarize    404          0.4960     0.3900     401
polarize    405          0.4960     0.3900     402
polarize    406          0.4960     0.3900     403
~
~
~
~
polar_renumbered.key                               1,1           All
```

Once renumbering is complete, we're ready to perform the final step of electrostatic parameter derivation - fitting of our raw multipoles to recreate the molecular electrostatic potential (MEP).

Multipole parameter assignment 5: Electrostatic potential fitting

We now have an initial set of multipoles for our ethanol molecule. These are already slightly modified from the original DMA multipoles as they now take into account the intramolecular polarisation from the AMOEBA induced dipole model, and have been averaged over chemically equivalent atoms. As such we'll now refit this set of multipoles to recreate a high-level QM electrostatic potential as best we can. For more complicated molecules we should also fit our multipoles to multiple different molecular conformations to ensure our parameters can accurately account for conformational changes in our MD simulation, but for ethanol a single conformational fit is likely sufficient.

For the fitting process we'll use a larger basis set than our geometry optimisation, incorporating diffuse and polarisation functions to generate a highly accurate MEP. A Gaussian input file has been provided, named `ethanol_esp.com`. The key line inside is:

```
#Sp MP2/aug-cc-pvtz Density=MP2 MaxDisk=10GB Geom=AllCheck
```

Indicating that this time we'll perform a single point calculation ('`sp`'), using the MP2 level of theory and the aug-cc-pVTZ basis set. All other options remain the same as before. A script has been prepared to run the MEP calculation, submit it with:

```
qsub MEP_calculation.x
```

This should again take around a minute or two to run.

Once complete, convert the binary checkpoint file into ASCII format again (type `formchk ethanol_esp.chk`), as we'll need the MEP to be readable by Tinker for the next steps. The multipole fitting process makes use of the Tinker `potential` program and consists of three steps:

1. For each conformation of interest, create a Cartesian grid of points at which to calculate electrostatic potential
2. Evaluate the QM electrostatic potential at each of these grid points, for each conformation
3. Evaluate the AMOEBA electrostatic potential at each of these grid points, for each conformation, and fit multipoles to minimize RMS error between QM & AMOEBA electrostatic potential across all conformations simultaneously

In this case we have only one conformation of interest, but the process is identical for multiple conformations - the multiple copies of important grid/potential/structure files can be concatenated together for use with `potential`.

Step 1:

Run the `potential` program interactively (type `potential` on the command line) and select option 1 to create a `cubegen` input file. When prompted, enter the name of your xyz file that has been *correctly renumbered* with atom types starting from 401 (`ethanol_renumbered.xyz`). Likewise enter the key file with correctly renumbered atom types (`ethanol_renumbered.key`).

As an output, `potential` will create a file of grid points around the molecular volume, `ethanol_renumbered.grid`, and suggest the next steps in the fitting process. We'll move on to those below.

Step 2:

As suggested by `potential`, the next step is to run Gaussian's `cubegen` program to calculate the QM electrostatic potential at the set of grid points we've just created. Run the following:

```
cubegen 0 potential=mp2 ethanol_esp.fchk ethanol_esp.cube -5 h <
ethanol_renumbered.grid
```

The resulting output file, `ethanol_esp.cube`, contains the QM electrostatic potential at the grid points defined in `ethanol_renumbered.grid`. Next, re-run the `potential` program, this time choosing option 2 to read in the `ethanol_esp.cube` file. This creates a Tinker-readable `ethanol_esp.pot` file - it is these files that can be concatenated if you wish to fit to multiple conformations simultaneously.

Step 3:

The final step is the fitting of the AMOEBA multipoles to recreate the QM electrostatic potential. So far all our multipole parameters have been contained in a key file, but as of Tinker 7, `potential` requires multipole parameters in a separate AMOEBA 'prm' file for the fitting step. We'll first have to create this prm file, which will contain a header defining various force-field-specific options (such as how to scale 1-4 nonbonded interactions, which form of vdW potential to use, etc.). You'll notice a file called `AMOEBA_header.prm` in your folder - these are the standard header options used for the AMOEBA force field. Combine this file with your current set of multipole parameters and you'll have a prm fil in the right format:

```
cat AMOEBA_header.prm ethanol_renumbered.key > AMOEBA_ethanol_prefit.prm
```

Next we still need to create a new key file to pass to potential, describing one additional option for the potential fit. Open up a new file, ethanol_potfit.key, and type the following:

```
parameters AMOEBA_ethanol_prefit.prm          # Read in parameters from prm file
fix-monopole                                  # Fix monopole values during
potential fit
```

We constrain the monopole values to their values derived from the DMA because the fitting process can cause them to drift from 'realistic' physical values - the monopoles do, after all, contribute the most to electrostatic potential and have the highest leverage in the fit.

Finally we're ready to run the potential fitting. Run potential interactively, but define our new keyfile on the command line:

```
potential -k ethanol_potfit.key
```

Then follow the instructions on screen, choosing option 6 to fit new electrostatic parameters, reading in the renumbered xyz file (`ethanol_renumbered.xyz`) and Tinker-readable QM `ethanol_esp.pot` file, and optimising parameters until a final RMS gradient of 0.1 (the default). Small molecules are usually safe to converge tightly to the QM potential, but larger molecules than ethanol may benefit from a looser minimisation criterion (e.g. 0.5). This would obtain nearly all of the benefits in terms of improved potential, without allowing the multipole parameters to drift far from the original DMA values.

The optimisation should take a few seconds, and should finish with a summary of the differences between AMOEBA and QM electrostatic potentials:

```
 Electrostatic Potential over all Grid Points :

 Average Potential Value for Model :                 0.0523
 Average Potential Magnitude for Model :             4.1551
 Average Potential Value for Target :                0.0791
 Average Potential Magnitude for Target :            4.1703
 Average Signed Potential Difference :              -0.0267
 Average Unsigned Potential Difference :             0.0553
 Root Mean Square Potential Difference :             0.0825
```

The final multipole parameters post-fitting will be printed out in the `ethanol_renumbered.key_2` file (the prefix is taken from the xyz you gave `potential`). Replace the old parameters in the `AMOEBA_ethanol_prefit.prm` file (perhaps renaming it at the same time) with the new ones, and our multipole parameter derivation is complete. You can find an example set of final multipoles in the `outputs` folder, labelled `AMOEBA_ethanol_postfit.prm`. In the next part, we'll assign valence and vdW parameters, test our parameters are stable, and set up for a series of free energy calculations.

註解