

# Tutorial for parametrization of the Drude polarizable force field  
# Fang-Yu Lin and Alex MacKerell  
# April 2017

## Introduction:

Parameter optimization protocols of small molecules in the Drude FF is similar to CGenFF. The target data is QM calculated geometries, vibrational spectra and energies, and experimental observables. (for further details see: <http://pubs.acs.org/doi/abs/10.1021/acs.chemrev.5b00505>)

The optimization procedure is as follows:

### ■ Electrostatics parameter optimization (Section 2):

Electrostatic parameters include charges, alpha, and Thole parameters. These are optimized targeting QM molecular dipole moment, molecular polarizability, and water-model compound interaction energies.

### ■ Bonded/internal parameters optimization (Section 3):

Bonded parameters include bond, valence angle, dihedral and improper torsion parameters; internal parameters are optimized targeting QM optimized geometry and vibrational spectra. If the crystal structures are available, the crystal bond and angle values are also considered.

Dihedral parameters may be subject to additional optimization targeting a QM potential energy surface, which is generated by rotating about the central bond in a dihedral.

### ■ LJ parameters optimization (Section 4):

In most cases, this is not needed, as the LJ parameters of the atom types corresponding to functional groups in the majority of compounds are already optimized. However, if needed, Rmin/2 and well-depth of LJ parameters for the atom type are optimized to reproduce the experimental enthalpy of vaporization and molecular volume of pure solvents, experimental enthalpy of sublimation and lattice parameters of crystals, rare gas interaction energies and hydration free energies.

## Notes:

- **The optimization requires iterative refinement for all stages due to the impact of, for example, changes in the internal parameters that alter the conformation and, therefore, the interactions with water.**
- **Please maintain the number of significant figures used for parameters to those used in the master toppar str file.**

---

## 1. Preparation of QM target data

Typically, we generate the Gaussian input files using CHARMM. This means that we need to generate the initial CHARMM topology and then generate the Gaussian input file based on this topology so that the atom ordering matches the toppar file. This is highly convenient for parametrization.

Example inputs described below are included for chlorobenzene (chlb, a planar system) and chloroethane (clet, a sp<sup>3</sup> tetrahedral molecule). These include examples of the use of lone pairs, but do NOT include the use of anisotropic polarizability. An example of that may be found for RESI NMA in the toppar\_drude\_master\_example.str file.

### 1.1 Generate the initial toppar file:

There are two ways to generate the initial toppar file (toppar\_drude\_example.str). However, both ways require careful checking to ensure the correct Drude atom types are correctly set.

(1) **Using ParamChem:** You may use ParamChem

(<https://cgenff.paramchem.org/commonFiles/simulationLinks.php>) to generate a preliminary stream file containing the initial charges and parameters in the context of the additive force field, and then convert that into Drude atom types based on analogy with Drude atom types already available. You also have to manually add the ALPHA and THOLE parameters as well as LONEPAIR<sup>1</sup> and ANISOTROPY parameters if needed (eg. for hydrogen bond acceptors). When building the toppar file manually, always refer to the main Drude toppar files.

(2) **Using GAAMP:** Alternatively you can employ GAAMP (<http://gaamp.lcrc.anl.gov/>), using passphrase **A journey to parameterization**) to generate all the parameters for your molecule. However, the atom types generated by GAAMP are CGenFF atom types, such that you will still have to convert them into Drude atom types and ensure they match the Drude toppar files. GAAMP is based on the **fitcharge** module in CHARMM that optimizes the charges using perturbed QM electrostatic potential maps. From GAAMP, you will receive an email containing a tgz file (this may take a few days depending on the size of the molecule). In the drude-result directory is a file report-esp.txt that contains the information you need to get started in CHARMM. (for further details see: <http://pubs.acs.org/doi/abs/10.1021/ct4003477>)

- **Note:** Lone pairs have been generated for a number of types of hydrogen bond acceptors, allowing for lone pair geometries to be directly transferred from the previously optimized molecules. However, if analogous functional groups aren't available; the placement of lone pairs may be performed as previously described<sup>1</sup>, including as part of the reproduction of water (or ion)-model compound interactions.

<sup>1</sup>Harder, E.; Anisimov, V. M.; Vorobyov, I. V.; Lopes, P. E. M.; Noskov, S. Y.; MacKerell, A. D.; Roux, B. J. *Chem. Theory Comput.* **2006**, 2 (6), 1587 (<http://pubs.acs.org/doi/abs/10.1021/ct600180x>).

### 1.2. Generate the QM geometry optimization input file based on the model compound RTF in the Drude toppar file using CHARMM:

Building the molecule in CHARMM can be done with the 0\_build\_<residue>.inp input file (<residue> is clet or chlb). In fact, you can use the same inputs as in CGenFF parametrization by updating the generate line for your residue and adding the commands in bold below to create the Drudes and lone pairs:

```
generate @residue first none last none setup warn drude dmass 0.4
....
coor sdrude
```

### *coor shake*

The input file (0\_build\_<residue>.inp) will generate a QM input that can be run on a local cluster. It is already set but make sure the QM input uses MP2/6-31G(d) (or higher level for sulfurs or halogens). It is necessary to delete the lines corresponding to Drudes and lone pairs (D and L lines respectively) from the QM input. Older versions of CHARMM may give strange characters when writing Gaussian input files so proceed with caution if an old version is used. For example, it may incorrectly write CL as C; you need to correct the atom name in the QM input.

**Example CHARMM inputs:** 0\_build\_chlb.inp (for chlorobenzene); 0\_build\_clet.inp (for chloroethane)

**Example QM input:** ./gauss\_input/<residue>\_opt\_freq\_mp2.gjf

#### **1.2.1. Extract the optimized geometry:**

**Example directory:** ./gauss\_opt

**Usage:** ./mp2geom <residue>\_opt\_freq\_mp2.log

**Example command:** ./mp2geom clet\_opt\_freq\_mp2.log  
./mp2geom chlb\_opt\_freq\_mp2.log

#### **Note:**

- **mp2geom** will call another script **recover03symm** in the same folder.
- **Prepare a <residue>\_min.crd** and put it in the same folder, as mp2geom will use it to generate <residue>\_mp2.crd. This coordinate file is generated by **0\_build\_<residue>.inp**; however, the lonepair and drude ("L" or "D") lines in <residue>\_min.crd should be removed, as <residue>\_min.crd is used as a template to build <residue>\_mp2.crd. Always check whether the format in <residue>\_mp2.crd is correct.
- **mp2geom** will also generate <residue>\_water\_mp2.xyz and <residue>\_opt\_freq\_mp2out.xyz that will be used in generating vibrational spectra below in section 1.3.1.

### **1.3 Generate the QM frequency input file**

**Example directory:** ./vibration\_spectra

The input files to obtain QM vibrational spectra are created by making a CHARMM input file (<residue>\_molvib\_qm.inp) that calls CHARMM's MOLVIB facility. The input file includes MP2 Cartesian (**CART**) coordinates of the molecule (copied from <residue>\_opt\_freq\_mp2.log) and the mass of each atom; a table of Internal Coordinates (**IC**); a U MATRIX (**UMAT**) that transforms these internal coordinates to Pulay et al.'s\* "local internal valence coordinates"; a Potential Energy Distribution (**PED**) section consisting of names for the local internal valence coordinates; the MP2 Force constant Matrix (**FMAT**) in Cartesian coordinates (copied from <residue>\_opt\_freq\_mp2.log), and a scaling factor that is consistently

applied to the force constant matrix at the corresponding optimized level of theory. Running **<residue>\_molvib\_qm.inp** will output the QM vibrational spectra.

\*P. Pulay, G. Fogarasi, F. Pang, J. E. Boggs, *J. Am. Chem. Soc.* **1979**, 101, 2550-2560 (<http://pubs.acs.org/doi/abs/10.1021/ja00504a009>).

- **Note:** The QM calculations for Drude parametrization were typically performed at the MP2/6-31G(d) level with scale factor of **0.9434** applied to vibrational modes to account for limitations in the level of theory. You may refer to <http://cccbdb.nist.gov/vibscalejust.asp> for the scale factor for higher levels of theory.

To quickly build **<residue>\_molvib\_qm.inp**, a stream file for MM vibrational spectra (**molvib\_mm.str**) needs to be built first, which will be used as a template to build **<residue>\_molvib\_qm.inp**. The following subsections illustrate how to build **molvib\_mm.str** and **<residue>\_molvib\_qm.inp**.

### 1.3.1: Generate **molvib\_mm.str**:

**Example directory:** ./vibration\_spectra/clet\_molvib

**Example directory:** ./vibration\_spectra/chlb\_molvib

This is to generate a stream file (**molvib\_mm.str**) to be used in a CHARMM input to calculate MM vibrational spectra by the MOLVIB facility in CHARMM. The script **molvib\_mm.py** can be used to generate **molvib\_mm.str**. However, first you need to build a template toppar (**toppar\_template\_chlb.str**) to run **molvib\_mm.py**. In this template topology for your molecule, remove lone pairs, LONEPAIR and ANISOTROPY information; the BOND information involving lonepairs, and blank lines between the ATOM and BOND information; otherwise, this information will lead to the script not working. (Please compare **toppar\_template\_chlb.str** and **toppar\_drude\_example.str** for more details)

**Scripts involved:** **molvib\_mm.py**; **ringsys**

**Usage:**

**./molvib\_mm.py <residue> <topology\_keyword> [auxiliary\_arguments]**

where:

- ♦ **<residue>**: eg. **chlb** (lower cases)
- ♦ **<topology\_keyword>**: Your template toppar file, eg. **toppar\_template\_chlb.str** (removing LONEPAIR and ANISOTROPY information and the BOND information involving lone pairs)
- ♦ If the molecule has rings, for each ring you need to add a line like "**!RING planar 6 C1 C2 C3 C4 C5 C6**" or "**!RING planar 5 C1 C2 C3 C4 C5**" in the template toppar file.
- ♦ **<auxiliary\_arguments>**: optional arguments for special cases; This script judges the symmetry type of an atom according to the number of connected hydrogen atoms. If an atom is connected to several atoms, among which none is hydrogen, you need to manually define the symmetry type. For example:
  - ♦ "**sp3 C:S1 H1:O1 H2:O2 H3:O3 X:C2**", which means that 'S1' is

the central atom, 'O1', 'O2', and 'O3' function like three hydrogens, and 'C2' functions like a 'X' atom. Please see Figure 1 in P. Pulay, G. Fogarasi, F. Pang, J. E. Boggs, J. Am. Chem. Soc. 1979, 101, 2550-2560 (<http://pubs.acs.org/doi/abs/10.1021/ja00504a009>).

- ♦ For atoms connected to 3 atoms, and none of which is hydrogen. Then supply an argument like "**sp2 C:<atom0> H1:<atom1> H2:<atom2> X:<atom3>**".
- ♦ If you are not sure of the symmetry type, the script will still output suggestions to assign the keywords.
- ♦ If your molecule has a ring structure, this script would **call another script "ringsys"**.
- ♦ **Double check** the PED section in molvib\_mm.str.
- ♦ In molvib\_mm.py, you may ignore additional usage information for CGenFF toppar files in the script; those commands should not be used for Drude parametrization.

**Outputs:** molvib\_mm.str

**Example command for chlorobenzene:**

```
./molvib_mm.py chlb toppar_template_chlb.str "sp2 C:C6 H1:CL6 X:C5 Y:C1"
```

If '**ringsys**' is put in another folder, put a key word of **ringsys** :

```
./molvib_mm.py chlb toppar_template_chlb.str "sp2 C:C6 H1:CL6 X:C5 Y:C1" "ringsys  
[directory]/ringsys"
```

**Example command for chloroethane:**

```
./molvib_mm.py clet toppar_template_clet.str
```

### 1.3.2: Generate <residue>\_molvib\_qm.inp.

This step is simpler once **molvib\_mm.str** is constructed. The script **molvib\_qm.py** is used to generate <residue>\_molvib\_qm.inp.

**Usage:**

```
./molvib_qm.py <residue>
```

**Note:**

- The directory to run **molvib\_qm.py** should have <residue>\_water\_mp2.xyz, <residue>\_opt\_freq\_mp2.log' files and **molvib\_mm.str**. The former two files are generated by **mp2geom** in Section 1.2.1.
- After running this script, you will get a file <residue>\_molvib\_qm.inp. You can run it directly with CHARMM.

**Example command:**

```
./molvib_qm.py chlb
```

```
./molvib_qm.py clet
```

**Outputs:** chlb\_molvib\_qm.inp; clet\_molvib\_qm.inp

## 1.4. Single point energy calculation based on the gas phase optimized structure:

For calculations of interaction energies with water (see below), it is necessary to calculate the monomer energy at the standard QM level of theory used for the Drude FF (MP2/cc-pVQZ). After the optimization (section 1.2) is finished, extract the coordinates and use the optimized structure to do a single point calculation at the MP2/cc-pVQZ model chemistry. **An important note:** for Drude parametrization, it is crucial to use the MP2 level of theory in QM calculations, including for calculation of the dipole and polarizability. This is invoked in Gaussian using the keyword **density=current**. The log file will show that the population analysis was done using MP2. Otherwise, HF level of theory is used to calculate dipole (which is wrong). The keyword **"polar"** is to calculate polarizability. Note the use of RIMP2 (density fitting) treatment of electron correlation accessible in a number of QM codes is acceptable.

**Example QM input:** `./gauss_opt/single_point/<residue>_ccpvqz.gjf`

➤ **IMPORTANT:** From the Gaussian output obtain the “Exact polarizability” where the  $\alpha_{xx}, \alpha_{xy}, \alpha_{yy}, \alpha_{xz}, \alpha_{yz}, \alpha_{zz}$  molecular polarizability components in the tensor are in units of (a.u.)<sup>3</sup>. Multiply these by **0.1482** to compare with the CHARMM output which is in units of Å<sup>3</sup>. **IMPORTANT:** The calculation time and memory requirements increase rapidly with the size of the molecule.

### 1.5. Generate water-model compound interaction geometries based on the optimized model compound structure from 1.2:

The QM water-model compound interaction energies are used as target data to adjust the atomic charges. There are two ways to evaluate water interaction energies (described in section 2.2). Below are examples for setting up QM water-model compound interactions for the two different ways to calculate QM water-model compound interaction energies. Note that the QM water-model compound interaction energies are calculated at RIMP2/cc-pVQZ with BSSE correction included. As always, make sure that the water interaction geometries are reasonable by checking that a water is interacting with the atom of interest and no secondary interactions are present.

**a) To get the minimum water-model compound interaction energies** (similar to the protocol in CGenFF parametrization):

First, generate the Gaussian input file using the optimized model compound structure and run it. Then, you need to calculate the single point interaction energy at RIMP2/cc-pVQZ with BSSE correction included.

**Examples for Gaussian input for water-model compound interaction:**

`./gauss_water/minimum_water_interaction/chlorobenzene/chlb_water_mp2_H2.gjf`  
`./gauss_water/minimum_water_interaction/chloroethane /clet_water_mp2_H2.gjf`

**Examples for single point calculation with Psi4:**

`./gauss_water/minimum_water_interaction/chlorobenzene/bsse`  
`./gauss_water/minimum_water_interaction/ chloroethane /bsse`

**b) To calculate the radial water interaction energy profile:**

The water-model compound radial scans involve varying the water to model compound distance while keeping the model compound at the gas phase optimized geometry and the water

at the SWM4 geometry. The water-model compound interaction geometry is built as a Z-matrix using Gaussian, and the Gaussian keyword "scan" can be used to do the potential energy scan based on the set variable (the variable in the example below is the distance between chlorine and oxygen of the water), which will generate each geometry based on changing the variable. Then, the single point calculation is performed on each generated geometry. The single point calculation is performed at RIMP2/CC-PVQZ with BSSE correction using PSI4. Details and scripts are in the respective folder.

**Examples for Gaussian input for water-model compound interaction:**

```
./gauss_water/radial_water_interaction/chlorobenzene/chlb_water_mp2_scan.gjf  
./ gauss_water/radial_water_interaction/chloroethane/clet_water_mp2_scan.gjf
```

**Examples for single point calculation with Psi4:**

```
./gauss_water/radial_water_interaction/chlorobenzene/bsse  
./ gauss_water/radial_water_interaction/chloroethane/bsse
```

**1.6. Generate potential energy surface by scanning a dihedral angle:**

a) You need to use the gas-phase optimized structure and build the input file for Gaussian.

**Example:** ./dihedral\_fitting/scan\_clcch21/\*.gjf

b) Once it finishes, do ./scan2cat\_gauss, then do ./c2s\_file to obtain the interaction energy as a function of dihedral angle.

**Usage:** scan2cat\_gauss <prefix for .c\_file and .e\_file> <gaussian log file>

**Usage:** c2s\_file <template crd> [<c\_file> <e\_file>]

➤ **Note:** "crd" folder should be generated and copy your <clet>\_<clcch21>.e\_file into scan\_<clet>.qme

---

## **2. Optimizing electrostatic parameters:**

Initial electrostatic parameters may be generated using GAAMP (section 1.1) or obtained by analogy to similar functional groups.

### **2.1. To reproduce the molecular polarizability:**

Manual adjustments of the ALPHA and THOLE parameters are usually necessary to reproduce the molecular polarizability. Then the charges are adjusted to reproduce water interaction energies and dipole moment. Although polarizability doesn't change a lot while the charges are changed you still have to check the values as part of the iterative parameter optimization process. If you are starting with results from GAAMP, QM molecular polarizability and dipole moment are often matched quite well.

a) The command in CHARMM to calculate molecular dipole moment:

```
coor dipole oxyz select segid @residue end  
?xdip ?ydip ?zdip ?rdip
```

b) The script to calculate molecular polarizability can be streamed in the CHARMM input:

## polar.efield.str

**Example CHARMM inputs:** 1\_polar\_dipole.inp

**Output:** polar.dipole.<residue>.dat

➤ **Note:** use 1\_polar\_dipole.inp to get the MM molecular polarizability (alphaxx+yy+zz). Typically the ALPHAs are scaled to be ~70-90% of the gas phase QM values (no scaling for alkanes or halogenated species), yielding QM molecular polarizabilities that are also ~70-90% of the gas phase QM values. Ideally, if pure solvent dielectric constants are available for the model compound, run MD and calculate the dielectric constants and use that as a guide for the ALPHA parameter scaling. When such data is not available using a **scaling factor of 0.85 for the ALPHAs**, which has been shown to work well for heterocycles and is intermediate to the range of values obtained for many model compounds, is suggested.

### 2.2. Reproduce water-model compound interaction energies and dipole moments:

Adjust the charges to optimize the agreement between the Drude and QM water -model compound interaction energies. Note that when the charges are adjusted, the dipole moments will change; thus, while optimizing the charges, you need iteratively check the dipole moment. There are two ways of fitting water-model compound interaction energies. One is to fit QM water-model compound interactions at the minimum energy geometry and the distance. The second is to fit to QM water-model compound interaction radial potential energy scans (ie. interaction energy as a function of distance). As always, make sure that the water-model compound interaction geometries are reasonable, meaning that a water is interacting with the atom of interest and no secondary interactions are present.

#### 2.2.1. Fitting the water-model compound interactions at the minimum energy geometry:

The Drude water-model compound interactions are done in a manner analogous to that in CGenFF (if you do not have experience in CGenFF parametrization, you may check the link at the bottom) by comparing the QM and MM minimum energy distances and geometries and adjusting the partial atomic charges to improve the level of agreement. Emphasis should be placed on the most favorable water-model compound interactions as they will dominate in MD simulations. **IMPORTANT:** For the Drude MM calculations, do not use the "inter" command as the contribution of the self-energy associated with relaxation of the Drude particles needs to be taken into account (as with the electronic degrees of freedom in a QM interaction energy calculation). This is performed by calculating the total energy of the water-model compound complex following relaxation of the Drudes and subtracting from that the sum of the water and model compound gas phase energies.

**Example CHARMM input:**

./drude\_water\_interaction/minimum\_water\_interaction/<residue>\_water\_mini.inp

#### 2.2.2. Fitting to QM water-model compound radial interaction energy scans:

This is to perform a radial scan by changing the distance between water and the model



compound with rigid, gas phase geometries.

**Example CHARMM input:**

```
./drude_water_interaction/radial_water_interaction/ <residue>_water.inp
```

(The Drude water-model compound interaction energy will be written in \*.plt. The first column in \*.plt is the distance; 2<sup>nd</sup> column is the interaction energies; 3<sup>rd</sup> column is the electrostatic contribution; 4<sup>th</sup> column is the Lennard-Jones contribution.)

➤ **Note:** Make sure that the Z-matrix in the CHARMM input is the same as the corresponding z-matrix in Gaussian.gjf

➤ Anisotropic polarizability:

One way to fit the anisotropic polarizability is to place an ion around the hydrogen bond acceptor (eg. oxygen) as a function of angle and dihedral, and adjust the anisotropic parameters to reproduce the QM interaction energies with the ion as a function of orientation. The ion could be  $K^+$  or  $Na^+$ , serving as a point charge to generate the electric field induce polarization of the targeted atom in the molecule. The following example is for a  $K^+$  placed around the oxygen of NMA. The angle of  $C=O...K^+$  ranges from  $90^\circ$  to  $180^\circ$  with an increment of  $15^\circ$  while the dihedral of  $N-C=O...K^+$  ranges from  $-165^\circ$  to  $-180^\circ$  at the fixed distance of  $O...K^+ = 2.6\text{\AA}$ .

The concept of the following procedure is: 1) build the NMA and ion geometries (nma.pot.2.6\_<ang>\_<dihedral>.crd) based on the additive force field, followed by converting it to Drude atom name (if atom names are different); 2) Generate QM input files (Qchem/Gaussian/Tinker input files) based on those \*.crd; 3) Similarly, the interaction energies obtained from MM will be calculated based on those \*.crd; 4) Extract the QM and MM interaction energies for comparison and adjust the anisotropic parameters to have better agreement.

**Example directory:** ./aniso\_nma

**Example commands:**

1) Charmm script gen.inp to generate coordinates:

```
charmm < gen.inp > gen.out
```

➤ **Note:** convert to the correct atom name using:

```
./makedrudecrd
```

(It is written for NMA and needs to be changed for other molecules)

2) Python script **aniso.py** to generate QChem/Gaussian input files based on the generated \*.crd in step 1). (The python script will create QChem inputs for nma-sodium geometries and Gaussian inputs for nma-potassium geometries). The Tinker xyz files may also be generated which are for AMOEBA and can be ignored here.

```
python aniso.py
```

➤ **NOTE:** If you need the Tinker xyz, the coordinate of xyz in aniso.py should be

changed to your molecule.

3) CHARMM script **drudeenergy.inp** to compute Drude interaction energy based on the generated \*.crd in step 1). The interaction energies are written into drude.nma.pot.2.6.ene.

```
charmm < drudeenergy.inp > drudeenergy.out
```

**Other scripts:** processqchem.py for extracting interaction energies from QChem output.

---

### 3. Optimizing bonded/internal parameters:

In this stage, the internal parameters will be optimized based on the agreement between the target and MM geometries. Once changes in the parameters are made, be sure to check the ability of the model to reproduce the remainder of the target data in an iterative fashion.

#### 3.1. Bonded/internal parameters:

The bond and valence angles are targeting QM optimized geometries and the crystal structures (if available, with crystal survey data being preferable). The force constants are targeting QM vibrational frequencies and assignments (the second part in the optimization Gaussian \*.gjf will provide frequency). Reproduction of force constants may also be used to optimize dihedral and improper parameters, with selected dihedrals optimized based on potential energy scans (section 3.2).

**Example directory:**

```
./quick_bond_angle/clet
```

```
./quick_bond_angle/chlb
```

The script **quick\_qm.py** can be used to extract QM bond and angle values. Note that the <residue>\_opt\_freq\_mp2out.xyz and <residue>\_water\_mp2.xyz should be in the same directory, which are generated by generated by mp2geom in Section 1.2.1.

**Usage:** ./quick\_qm.py <residue>

A stream file called **quick\_qm\_<residue>.str** will be created that provides all the bond and angle information of the molecule. Copy it into another file called **quick\_qm\_mm<residue>.str**. Then assign which bond and angles are to be measured using the **quick** command. Only the bond and angle parameters that are **NOT** in the main toppar files need to be optimized here as these parameters have been previously optimized targeting other molecules and cannot be changed (See: quick\_qm\_clet.str, quick\_qm\_mm\_clet.str, quick\_qm\_chlb.str, quick\_qm\_mm\_chlb.str; the **quick\_qm\_mm<residue>.str** will be read in CHARMM input: **quick\_bond\_angle.inp** ).

**Usage:** charmm < **quick\_bond\_angle.inp** > **quick\_bond\_angle.out**  
(The MM bond and angle will be printed out in **quick\_bond\_angle.out**)

**Note:**

➤ The atom IDs will be different than those in <residue>\_mp2.crd extracted from the

Gaussian optimized structure, as the Drude particle are included in the molecular coordinates when you run CHARMM; thus, be careful to assign the correct atoms for the quick command.

➤ For angles involved in a planar structure, the sum of the angle under parametrization (eg.  $A1^\circ$ ) together with other angles (eg.  $A2^\circ$  and  $A3^\circ$ ) should follow  $A1^\circ + A2^\circ + A3^\circ = 360^\circ$  around a central carbon. Similarly, the sum of the internal angles in a planar 5-member ring should equal  $540^\circ$  and those in a planar 6-membered ring should equal  $720^\circ$ .

a) Get MM frequency:

The script `quick_bond_angle.inp` will also read the stream `molvib_mm.str` (described in section 1.3.1) to obtain MM frequency.

**Usage:**

```
charmm < quick_bond_angle.inp > quick_bond_angle.out
./molvib_energy quick_bond_angle.out > mm_table
```

b) Get QM frequency:

The `<residue>_molvib_qm.inp` (described in section 1.3.2) is used here.

**Usage:**

```
charmm < [residue]_molvib_qm.inp > [residue]_molvib_qm.out
./molvib_energy [residue]_molvib_qm.out > qm_table
```

Finally, compare `qm_table` and `mm_table` to see whether QM and MM vibration spectra are in good agreement.

➤ Reference: P. Pulay, G. Fogarasi, F. Pang, J. E. Boggs, *J. Am. Chem. Soc.* **1979**, 101, 2550-2560. (<http://pubs.acs.org/doi/abs/10.1021/ja00504a009>)

### 3.2. Dihedral parameters:

Once you prepare the QM folder (ex: `scan_clcch21`), create another folder parallel to the QM folder (ex: `fitting_verify`), then calculate the CHARMM generated dihedral potential energy scans. Note that dihedral parameters that are based only on non-hydrogen atoms about rotatable bonds are typically the only ones optimized based on PES. An exception includes hydroxyl groups.

**Example CHARMM input:**

```
./dihedral_fitting/dihe1/fitting_verify/allscan_drude.inp
./dihedral_fitting/dihe2/fitting_verify/allscan_drude.inp
Please see detailed procedures in:
./dihedral_fitting/dihe1/fitting_verify/readme_verify
./dihedral_fitting/dihe2/fitting_verify/readme_verify
```

**Usage:**

1. `./initialize`
2. `charmm < allscan_drude.inp > allscan_drude.out`
3. `./scanplot -i [output_plot_name] scan_[residue].qme scan_[residue].mme`

**Example commands:**

```
./initialize  
charmm < allscan_drude.inp > allscan_drude.out  
./scanplot -i test scan_chlb.qme scan_chlb.mme
```

For multiple dihedrals, the dihedral parameters can be fit using the lsfitpar program (see Section S2.)

---

#### **4. Optimizing LJ parameters (this step is not necessary if the atom type already exists which will generally be the case for drug-like molecules):**

##### **4.1. LJ parameter (Rmin/2 and well-depth(epsilon)) optimization**

LJ parameters (Rmin and the well depth (epsilon)) are optimized targeting bulk phase simulations and gas phase simulations to reproduce the experimental heats of vaporization (delH) and molar volume (Vm). Prepare a box of pure solvent to run the neat simulations and extract a coordinate file for the gas phase of your molecule from previous steps. Alternatively, LJ parameters may be optimized targeting crystal data and interactions with rare gas atoms can be used to facilitate the optimization process (The way to build QM interaction inputs with rare gas atoms is similar to that for water interaction, see **example**: ./rare\_gas. Note that the QM calculation is performed at the MP3/6-311++G(3d,3p) level of theory for each rare gas atom orientation, reference: [Yin D, MacKerell AD., Jr. \*J Comp Chem.\* \*\*1998\*\*;19:334–348](http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1096-987X(199802)19:3%3C334::AID-JCC7%3E3.0.CO;2-U/abstract) ([http://onlinelibrary.wiley.com/doi/10.1002/\(SICI\)1096-987X\(199802\)19:3%3C334::AID-JCC7%3E3.0.CO;2-U/abstract](http://onlinelibrary.wiley.com/doi/10.1002/(SICI)1096-987X(199802)19:3%3C334::AID-JCC7%3E3.0.CO;2-U/abstract)).

**Example:** ./vdw\_parameters/hval\_vm

**Example for bulk simulation:** ./vdw\_parameters/hval\_vm/box.inp;

**Example for gas phase simulation:** ./vdw\_parameters/hval\_vm/one.inp

**Example for gas phase simulation based on the geometry of each molecule in the bulk simulation:** ./vdw\_parameters/hval\_vm/gas\_2/qsub\_random.sh;

**Example for generating a box of molecules:** ./vdw\_parameters/hval\_vm/initbox.inp

- **Note:** The delH, Vm, and dielectric constant can be simulated based on box.inp (be sure to change the temperature so it corresponds to experiment).

Here are scripts to calculate those values (under ./vdw\_parameters/hval\_vm ):

- hval.py (delH)
- vm.py (Vm)
- enma.py (dielectric constant)

- **Note:** usage instructions are written in the respective \*.py and that the **enmf** in **enma.py** should be assigned based on the Clausius-Mossotti equation (see the brief description <http://pubmedcentralcanada.ca/pmc/articles/PMC2542883/>)

##### **4.2. Free energy of aqueous solvation:**

**Example:** ./vdw\_parameters/fep/

Simulate 1) the model compound in a water box; 2) in the gas phase; 3) with additional simulations for calculating the Lennard-Jones long-range correction.

The files for 1) and 2) are similar. You need to modify parset.str and do **./fep.run**. Once the simulations are finished, do **./xtract.run** to get the energies.

Reference for FEP calculation: <http://pubs.acs.org/doi/abs/10.1021/jp048502c>

---

## S1. MCSA

MCSA (Monte Carlo Simulated Annealing) can be used to optimize the charges, alphas, and Thole parameters. Once this is done, use the converged parameters and stream those in the above CHARMM scripts to check the ability to reproduce the various target data and reoptimize as required. Note that calculating average values over the last 10 MCSA steps once the run has converged will minimize noise associated with the small variations in the parameter occurring due to the MC steps. Please maintain the number of significant figures used for parameters as in the master toppar str file.

**Example:** ./mcsa/

**- Main script:**

sannl-dipole-polar.f

**- Things to prepare/modify:**

1. saparams-dipole-polar.prm # Include detailed explanation below
2. stream.txt # this will generate the temporary para.str to be read in chlb\_msca.inp
3. chlb\_msca.inp # charmm input
4. toppar\_drude\_mcsa\_template.str  
# sannl-dipole-polar.f will generate a 'para.str' based on stream.txt; 'para.str' will be read in toppar\_drude\_mcsa\_template.str, so this template should include variables corresponding to the variables in stream.txt (you may compare toppar\_drude\_mcsa\_template.str and toppar\_drude\_example.str for clarification )
5. script-charmm-dip-alpha # script to call charmm input
6. dipoleqm.dat # access the qm dipole moment or molecular polarizability
7. polar.efield.str # Don't need to change this
8. create a folder named: OPTIMA # the resulting parameters are saved here.
9. chlb\_mp2.crd

**- Command:**

1. compile fortran: ./compile.sh
2. ./sannl-dipole-polar.x > result.dat

**Note:**

➤ Check chlb\_msca.out to see if there is error. Sometimes the incorrect order or variables

in stream.txt, or the initially incorrect total charge in saparams-dipole-polar.prm will cause problems.

- Remember to compile the fortran code after you edit stream.txt and saparams-dipole-polar.prm.
- More target data may be added (water interaction and so on) by editing the code ( sannl-dipole-polar.f). Then compile the fortran code and rerun.
- MCSA will be more efficient if your initial parameters are close to the optimized values; then you only need to assign smaller range for sampling to find better parameters.
- Limiting the range of sampling is important to assure that large force constants for the dihedral parameters that can cancel each other are not obtained. And it is strongly suggested that only multiplicities of 1, 2, 3 and 6 and phases of 0 and 180° be used.

---

## S2. lsfitpar program.

See <https://github.com/kennovo/lsfitpar> for detailed instructions.

**Example directory:** ./dihedral\_fitting

There are two examples (dihe1 and dihe2), both examples include two new dihedrals in the toppar file:

```
"CD2R6A   CD2R6A   CD2R6A   CLDR1" corresponds to c4-c5-c6-cl dihedral in CHLB
"HDR6A    CD2R6A   CD2R6A   CLDR1" corresponds to h5-c5-c6-cl dihedral in CHLB
```

Also, dihe1 and dihe2 both include examples of "autofitting" and "fitting\_verify", and the **lsfitpar** is performed in autofittig folder. The "fitting\_verify" is to use the predicted parameters and do the potential scan for verifying the predicted values (as described in Section 3.2)

- The autofitting in dihe1 gives the example of using lsfitpar to fit only "**CD2R6A CD2R6A CD2R6A CLDR1**" while keeping "**HDR6A CD2R6A CD2R6A CLDR1**" fixed at the initial guess.
- The autofitting in dihe2 gives the example of using lsfitpar to fit both "**CD2R6A CD2R6A CD2R6A CLDR1**" and "**HDR6A CD2R6A CD2R6A CLDR1**".
- The different files under 'autofitting' folder in "dihe1" and "dihe2" are **toppar\_drude\_autofitting.str** for CHARMM and **chlb\_lsfitpar.in** submitting to lsfitpar. Noted that the force constants of the dihedral parameters to be predicted from lsfitpar should be set to zero; thus **toppar\_drude\_autofitting.str** in dihe1 is only set to zero for "**CD2R6A CD2R6A CD2R6A CLDR1**" and the other "**HDR6A CD2R6A CD2R6A CLDR1**" is not changed. Also, **chlb\_lsfitpar.in** in dihe1 will assign one dihedrals for fitting and give one predicted dihedral parameters, while **chlb\_lsfitpar.in** in dihe2 will assign two dihedrals for fitting and give two predicted dihedral parameters.

The procedures for running lsfitpar are same in dihe1 and dihe2. The following example procedure is based on dihe1:

- 1. Create a toppar file for auto fitting.** The dihedral parameters to be estimated in this toppar file should set to zero (see toppar\_drude\_autofitting.str)
- 2. Prepare an "initialize" file.** It is just a bash script to create files where CHARMM will

output the values when running " allscan\_drude\_lsfitpar.inp".

**3. Prepare "allscan\_drude\_lsfitpar.inp".** Parameters are described through the comments in the file.

**4. Do 'initialize', then run CHARMM:**

```
./initialize
```

```
charmm < allscan_drude_lsfitpar.inp > allscan_drude_lsfitpar.out
```

- It will also generate ini\_geom.pdb (qm) and fin\_geom.pdb (mm) which can be viewed in VMD. The former is the QM dihedral scans; the latter is the MM dihedral scans

**5. Run lsfitpar:**

```
./lsfitpar -i [output_name].prm [output_name].ene
```

- **Note:** lsfitpar will use an interactive input, if you want to run it in background, you may prepare those input (eg. chlb\_lsfitpar.in) and do :

```
./lsfitpar -i [output_name].prm [output_name].ene < chlb_lsfitpar.in
```

- It is better to use the interactive input for the first time. The interactive input will give explanations while asking for input for the keywords.
- The scan\_chlb.qme in chlb\_lsfitpar.in is from the \*.e\_file (eg. cat ../scan\_h5c5c6cl/chlb\_h5c5c6cl.e\_file > scan\_chlb.qme)
- If you have more dihedral scans, all of them should be copied in the \*.qme and in the same order as you assign 'peses' in 'allscan\_drude\_lsfitpar.inp'.

### **Brief usage:**

1. ./initialize

2. charmm < allscan\_drude\_lsfitpar.inp > allscan\_drude\_lsfitpar.out

3. ./lsfitpar -i [output\_name].prm [output\_name].ene < chlb\_lsfitpar.in

- [output\_name].prm includes the estimated parameters for dihedral
- [output\_name].ene includes the aligned energy from QM, MM0 (without fitting) MME (using the estimated parameters)

4. ./fitplot -i [output\_plot\_name] [output\_name].ene

- It will generate all[output\_plot\_name]\*.gpt, which can generate the plot again.

5. Verify the [output\_name].prm. Copy them in your stream file and verify the dihedral PES as in fitting\_verify folder (See section 3.2).

### **Example commands:**

```
./initialize
```

```
charmm < allscan_drude_lsfitpar.inp > allscan_drude_lsfitpar.out
```

```
./lsfitpar -i chlb.prm chlb.ene < chlb_lsfitpar.in
```

```
./fitplot -i test chlb.ene
```

---

### **Final Consideration of the parameters themselves.**

- Always check that final parameters are physically reasonable by comparing to

parameters from similar functional groups. This is important as parameter correlation is a significant issue that can lead to physically unrealistic parameters.

- Dihedral parameters can be treated with a Fourier series of parameters with different multiplicities. It is strongly suggested that only 1, 2, 3 and 6 fold multiplicities be used. If after optimization the force constants for one of the multiplicities is small, omit it and redo the fitting.
- The phase of dihedral parameters should always be 0 or 180° as this assures that energy associated with changes in chirality are properly treated. CHARMM can use any value for the phase, but non 0/180 values are rarely needed.
- And always rerun all the calculations with the final set of parameters to be sure that the results are in acceptable agreement with all the target data.

---

**Additional information:** CGenFF tutorial at: <http://dogmans.umaryland.edu/~kenno/tutorial/>

---

**If you have questions, please contact:**

fylin[AT]outerbanks.umaryland.edu; alex[AT]outerbanks.umaryland.edu