Supporting information for "Hyperpolarizabilities of extended molecular mechanical system: an evaluation of classical models"

Ignat Harczuk,* Olav Vahtras, and Hans Ågren

KTH Royal Institute of Technology, School of Biotechnology, Division of Theoretical Chemistry and Biology, SE-106 91 Stockholm, Sweden

E-mail: harczuk@kth.se

Detailed computational protocol

Here the computational procedures are described, starting with the calculation using the DALTON quantum chemistry program, followed by the LoProp python script used to obtain LoProp properties and finally the script to obtain the Applequist interaction response.

A complete guide on how to perform these calculations can be found in the form of an Jupyter notebook[1], where a demonstration for a model system can be followed step-by-step interactively. Another way to perform these calculations, which hides the underlying complexity, is demonstrated in the Jupyter notebook[2].

All below procedures can be run on a generic Linux system.

[1] https://github.com/fishstamp82/moltools/blob/master/share/tutorial/two_water_example.ipynb

[2] https://github.com/fishstamp82/moltools/blob/master/share/tutorial/moltools_tutorial.ipynb

DALTON

The DALTON program can be obtained from www.daltonprogram.org. As of today, January 21, 2016, it has a custom license which requires the user to sign a user agreement. It is free to use with the above restrictions. Once a standard DALTON installation is obtained, the accompanying run-script should be ran as

```
>>> dalton -get 'AOPROPER AOONEINT' input.dal molecule.mol
```

The input.dal tells DALTON what calculations to perform and is listed in minimal form below using TDHF. The molecule.mol specifies the system to be evaluated, together with basis, charges, symmetries, and units to use. An example of molecule.mol is given below necessary to run the calculation on a water molecule. See the dalton script documentation for extended use.

input.dal

```
**DALTON INPUT
.RUN RESP
*END OF GENERAL
**WAVE FUNCTION
.INTERFACE
.HF
**INTEGRAL
.NOSUP
.DIPLEN
.SECMOM
**RESPONSE
*LINEAR
.DIPLEN
*END OF
```

molecule.mol

```
ATOMBASIS
comment 1
comment 2
Atomtypes=2 Charge=0 Nosymm
Charge=8.0 Atoms=1 Basis=ano-1 4 3 1 0
O 0.00000 0.00000 0.00000
Charge=1.0 Atoms=2 Basis=ano-1 2 0 0 0
H 1.43043 0.00000 1.10716
H -1.43043 0.00000 1.10716
```

LoProp

LoProp exists as a public repository and can be obtained from https://github.com/vahtras/loprop. It is written in python 2.7 and needs the binaries obtained from dalton. The output from the above DALTON calculation

```
>>> dalton -get ’AOPROPER AOONEINT’ input.dal molecule.mol
```

will produce an archive input_molecule.tar.gz file, which is simply used by LoProp to make the transformation of the molecular properties to a per-atom basis. Running

```
>>> loprop.py -l 1 -a 1 -f input_molecule.tar.gz
```

will output the atomic charges, and dipoles, together with the isotropic polarizabilities, of each atom in the order they are listed in the molecule.mol file.

Applequist

The scripts used to set up interacting particles can be obtained from https://github.com/vahtras/pd. In this demonstration, we use the particles script. To obtain the Applequist α^{Cluster} for two water molecules at 5 Å distance separation in the Z direction, using the TDHF method from above, we can run the blow python script. The format of input to the pd module is the same as for QMMM ṗot files for DALTON, which can be found in the DALTON manual. Using charges, dipoles, and isotropic polarizabilities, an example python script below is given. Note that since the hyperpolarizability is here zero, the polarizability will not depend on the local field of each particle.

```
#!/usr/bin/env python

from particles import *

h2o_string = """AU
6 1 1 0
1 0.000 0.000 0.000 -0.664 0.000 0.000 0.345 3.759
1 1.430 0.000 1.107 0.332 -0.161 0.000 -0.113 1.124
1 -1.430 0.000 1.107 0.332 0.161 0.000 -0.113 1.124
2 0.000 0.000 5.000 -0.664 0.000 0.000 0.345 3.759
2 1.430 0.000 6.107 0.332 -0.161 0.000 -0.113 1.124
2 -1.430 0.000 6.107 0.332 0.161 0.000 -0.113 1.124"""

pdl = PointDipoleList.from_string( h2o_string )
print pdl.alpha()
```

Note that the output from loprop.py has to be modified in order to produce two separate units. All lines starting with the same integer group will belong to the same residue, which can not polarize within itself, and thus only feels the field from other groups.

For calculations that automate all the above steps, see the tutorial provided in URL above, which can perform the DALTON step, LoProp, and generation of interacting systems automatically.

Dimer System

Basis set chosen

The basis set chosen is the equivalent of 6–31+G* using an ANO basis, which is commonly employed in computation of polarizabilities of larger molecules as it gives a good trade-off between accuracy and computational time. For our purposes, this serves as to deepen the qualitative understanding of the presented quadratic Applequist scheme, and need not at this point be qualitatively scrutinized.

Rotation

The rotation was done by defining the new set of basis vectors with the dipole moment of the water molecules being the positive z-axis. The water molecules plane would be in the xz-plane. This would lead to a positive x-axis in one of the hydrogen bond directions.

The rotation matrix transforming the new basis-vectors to the canonical basis vectors can be written

$$R = (e_x^{'}, e_y^{'}, e_z^{'}) \tag{1}$$

The inverse, or transpose of R, could then be used to obtain the rotated coordinates for an arbitrary water rotation from the template water which had the dipole in the z-axis, and hydrogens in the zx-plane as

$$p_i = R_{ia}^{-1} p_a \tag{2}$$

$$\alpha_{ij} = R_{ia}^{-1} R_{jb}^{-1} \alpha_{ab} \tag{3}$$

$$\beta_{ijk} = R_{ia}^{-1} R_{jb}^{-1} R_{kc}^{-1} \beta_{abc} \tag{4}$$

Properties for cluster

Since the SHAKE algorithm keeps bonds and the water fixed, we only calculate the molecular properties for one water molecule to serve as a template. It can be shown that the maximum angle and bond deviation is 1% yielding an error in properties that is around the same order, in the worst case. On average, however, the estimated error due to using the template method is one magnitude lower than the worst case scenario. Since each water molecule does not need a separate quantum mechanical calculation, this is a very good method to decrease the computational time. We thus only run the full quantum mechanical properties calculation on the different configurations of the cis-pinonic acids, as their conformation will change more drastically over time.

The attached pdb file "wat_500.pdb" has the full water configuration used in the extraction of clusters. The way all water clusters were chosen was to select the N first water molecules closest to the center of the box spanning from the minimum x, y, z coordinate to the maximum.

Properties

The rotationally invariant (isotropic) properties are defined as follows in Einstein notation

$$\mu_{iso} = \sqrt{\mu_i^2} \tag{5}$$

$$\alpha_{iso} = \frac{\alpha_{ii}}{3} \tag{6}$$

$$\beta_{||} = \frac{3}{5} \frac{\beta_{ijj}\mu_i}{||\boldsymbol{\mu}||} \tag{7}$$

Cube

The algorithm of constructing the cubes reads as follows

- Define a box length L
- Get the minimum and maximum coordinate points for a cluster, $\mathsf{p}_{\min} = (x_{\min}, y_{\min}, z_{\min})$ $\mathsf{p}_{\max} = (x_{\max}, y_{\max}, z_{\max})$
- Create a 3d cell with boxes B_{ijk} that have indices $i = C(\frac{\mathsf{p}_{\max,x}}{L})$, $j = C(\frac{\mathsf{p}_{\max,y}}{L})$, $k = C(\frac{\mathsf{p}_{\max,z}}{L})$, where C is the ceiling function
- Iterate over all point dipoles, placing them in box B_{ijk}, with indices of particle n obtained as $i = C(\frac{x_n - \mathsf{p}_{\min,x}}{L})$, $j = C(\frac{y_n - \mathsf{p}_{\min,y}}{L})$, $k = C(\frac{z_n - \mathsf{p}_{\min,z}}{L})$