COULOMB.py © 2012

Author:
Bartosz Błasiak
Wrocław University of Technology
`globula@o2.pl`

Contents

1 Introduction

This simple program is designed for calculation of coulomb interaction energies between two molecular individuals in terms of *ab initio* quantum mechanical methods at various levels of approximation. It serves the calculations using the following procedures listed:

1. Cube Method
2. Multipole Interaction Energy from Cumulative Atomic Multipole Moments Method (CAMM)
3. Charge Interaction from Electrostatic Potential Fitting Method (ESP)
4. First-order electrostatic interaction energy from Hybrid Variational-Perturbtional Interaction Energy Decomposition Scheme (EL(1) EDS)

The COULOMB.PY package also provides an opportunity to calculate coulomb transition energy couplings basing on external density matrices loaded from Gaussian output files.

The following tutorial is divided into three main sections (apart from this introduction). In the first the overall program structure is described. The detailed description of classes and functions is provided along with the further explanation of input file syntax which is very simple. Second section is devoted to rather very short and dirty introduction or summary[1] of the basic concepts of evaluation of coulomb interaction energy using different aproaches. This section is also a listing of what is actually implemented in current version of COULOMB.py and what is not. The last part is an appendix containing the review of auxiliary methods which are important tools for evaluation of electrostatic interaction energy.

2 Installation and program structure

The program uses modified version of PyQuante suite written in Python as well as external library for evaluation of two-electron integrals. The installation processes are the same as in the case of original packages. Probably to install the modified version of PyQuante you have to simply type the following command:

```
sudo python setup.py install
```

The program is divided into several files:

`coulomb.py+` contains main function `Main(argv)` with command line parser

`do.py+` contains the task selection routines in class `DO(PARSER)`

`parser.py+` contain a parser `PARSER` for input files

`run.py+` contains class `RUN` which is a base class for other classes representing methods like `ESP` and `MULTIP`. `RUN` represents the basic variables that are needed to perform other calculations for one molecular system (thus it gathers molecular data, multipole integrals and density matrix). More detailed description of this class is given in the text below.

`esp.py+` contains all procedures for performing ESP fitting

`multip.py+` contains all procedures needed for obtaining multipolar distributions of molecular species (either MMM or CAMM -see below).

[1]for those who are familiar with the subject

`eeleds.py+` contains all the procedures for calculations of electrostatic interaction energy in 1-st order from EDS. This class does not inherit from `RUN`.

`util.py+` contains some utility functions. The most important is `read_transition_dmatrix`. Others are timing processing utility `TIMER`, and print helpers.

2.1 Main classes

The most important classes are described below:

2.1.1 PARSER

This object contains some defaults which are assumed at the very begining during the input file reading. The defaults are for e.g. Angstroms as units for structure input and density matrix file type format (Gaussian) in case the user would provided an external source of this matrix. It contains also `self.transition=False` as well as `state=1`. The method `method` withdraws the following: method (like HF - in the case then the density matrix has to be calculated by using PyQuante routines so without providing external density matrix file), basis set and other keywords: `trans`, `units=[unit]`, `mpoints=[no of points per atom for ESP]`, `eint=[method(s)]`, `mtp=[method(s)]`, `pot=[method]`, `pad=number` and `print`. Method and basis set are provided in the input file similarly as in Gaussian input files (see section 2.2). Method `molecules` withdraws molecular coodrinates, multiplicity, charge and (optionally) density matrix. It looks for keywords `&MOL`, `DMATRIX=[file]` and molecular data. The constructor takes only the input file as a path.

2.1.2 DO(PARSER)

The obcject of this class has to find out that tasks to perform and to run appropriate routines. `self.mtp_methods` contains the list of multipole distribution tasks. Available are the two: Molecular Multipole Moments distribution, MMMs and Cumulative Atomic Multipole Moments, CAMMs. `self.eint_methods` contains the lsit of interaction energy calculation tasks. Available within this kind are interaction energy from CAMMs, ESP and EEL(1)EDS. Important information is that the `PARSER` stores finally a `Molecule` object from PyQuante routines (see class `Molecule` in file of PyQuante package) and lists of density external matrices. The constructor takes only the input file as a path.

2.1.3 RUN

This class contains memorials for molecular and method specific data. The constructor takes `molecule,basis,method,matrix=0,multInts=0`. If multints is set to zero the class `RUN` calculates density matrix (at HF level only) using PyQuante routines.

2.1.4 ESP(RUN)

Constructour takes the arguments:

```
def __init__(self, molecule, basis, method, mpot=1000, pot='CAMM', pad=10,
                   SVD=False, matrix=None, multInts=None, cnt=0.0100,
                   stat=False, Print=True, transition=False):
    RUN.__init__(self, molecule, basis, method, matrix, multInts)
    ...
```

The meaning of these arguments is as follows: mpot is a number of points *per atom* in which the potential has to be evaluated. The algorithm uses **random point selection**. pot specifies the method which potential will be evaluated from. It can be CAMM or WFN (wavefunction). pad is a box padding given in Bohrs, in which the points will be created. The centre of the box is assumed to be the geometrical centre of of the molecule given in molecule. cnt denotes condition number threshold being the limit number for assessing the real order of distance matrix (the analysis will be performed when SVD=True only[2]). Print statement says if the fitted charges are to be printed out and transition is to specify if we calculate the transition potential from transition density matrices (then nuclear contributions to the potential are zero).

2.1.5 MULTIP(RUN)

Constructeur takes the arguments:

```
def __init__(self, molecule, basis, method,matrix=None,multInts=None,
                   transition=False):
    RUN.__init__(self, molecule, basis, method,matrix,multInts)
    ...
```

The arguments are the same as in RUN class and one additional transition has exactly the same meaning as above described ESP class. For the purpose of the calculations for this class several small modifications were made in original PyQuante suite. One of this is adding multipole integrals formation (getM procedure) and adding a memorial LIST1 to instances of bfs class which indicates the number of atom as a function of atomic orbital and is an object of type list implemented in Python. For example

```
LIST1 - the list of atoms in the order of basis functions used,
e.g.: for h2o molecule with atoms: 8,1,1 and STO-3G basis
LIST1 = 0    0    0    0    0    1    2
        |    |    |    |    |    |    |
       1s   2s   2px  2py  2pz  1s   1s
```

The numbers as atomic indices are maintained in Python's convention for labeling list, array and string elements from 0 value. LIST1 memorial is needed when evaluating of CAMMs (see section 4).

2.1.6 EELEDS

The constructor takes the arguments as follows:

```
def __init__(self,molecule1,molecule2,basis,method,
             Transition=False,Exchange=False,
             matrixa=None,matrixb=None):
    ...
```

Statements molecule1 and molecule2 refer to two molecules, for which EEL(1)EDS will be calculated. Transition has the same meaning as previously. Exchange means exchange term from $\mathscr{K}_{i_1 j_1 k_2 l_2}$ integrals. matrixa and mtrixb (optional) are externam density matrices calculated in DCBS respectively for monomers molecule1 and molecule2.

[2]not yet written!

```
# CIS/6-31G* EINT=CAMM,ESP  TRANS mpoints=3200

&MOL1
ETHYLENE 1
DMATRIX=e1.cis-6-31Gd.log

0 1
C     -0.031871   -0.000007    0.001312
H     -0.053949   -0.102551    1.069139
C      1.188324    0.000003   -0.695389
H     -0.962724   -0.100044   -0.523396
H      1.209504   -0.099215   -1.763690
H      2.119240   -0.102685   -0.171502

&MOL
ETHYLENE 2
DMATRIX=e2.cis-6-31Gd.log

0 1
C     -0.031871    3.000007    0.001312
H     -0.053949    3.102551    1.069139
C      1.188324    2.999997   -0.695389
H     -0.962724    3.100044   -0.523396
H      1.209504    3.099215   -1.763690
H      2.119240    3.102685   -0.171502

&END
```

Figure 1: Input file for coulomb coupling constant evaluation for ethylene dimer obtained at CIS/6-31G* level of theory and using two methods: CAMM and ESP. Number of points per atom is set to 3200. Density matrices are provided for each the monomer. Units are assumed to be in Angstroms (default).

2.2 Input file syntax

In the input file one can provide more than one molecules to calculate multipole distributions. If one can obtain interction energy the only one amount of molecules given is two. The input file philosophy is similar for Gaussian and Molcas in order to make it more easy to handle. Beneath, an examplary input file is depicted. There are two parts of input file: methods etc in the form:

```
# [method]/[basis set] other commands
```

There cannot be any white character in the fragment `[method]/[basis set]`. Commands are as follows:

- `MTP=[method1],[method2],...` where methods are `CAMM` and `MMM`. Examples are `MTP=MMM` or `MTP=CAMM,MMM`.
- `EINT=[method1],[method2],...` similarly like above. Available methods are `MTP=ESP`,

MTP=CAMM and MTP=EELEDS. Example MTP=CAMM,ESP,EELEDS will compute interaction energy using three methods for the same pair of molecules. IMPORTANT! I didn't finished this part so for now the program will be calculatining molecular specific variables each time for each method separately so it is not so good (e.g. calculates multipole integrals twice for ESP (if potential has to be calculated from CAMM) and CAMM. I will solve this problem soon. I wanted to make class

- MPOINTS=[integer] specifies the number of point per atom
- PAD=[real] specifies molecular box padding for random points generation
- TRANS sets transition to be true (requires external transition density matrix - see below)
- EXCH sets exchange in EELEDS method for interaction energy computation
- UNITS=[unit] specifies units. Available are Angstroms (default) and Bohrs.

The second part of the input file specifies molecule(s) and has the form:

```
&MOL
[molecule name]
DMATRIX=[file with density matrix]

[charge] [multiplicity]
[atom_1]  [x] [y] [z]
[atom_2]  [x] [y] [z]
...
[atom_n]  [x] [y] [z]

/other molecules starting with new &MOL statement!/

&END
```

The line containing DMATRIX= is of course optional. All the input querries are case insensitive.

In order to use COULOMB.PY one has to simply load the input file in the following way:

```
coulomb.py [input_file]
```

After this computations start.

3 Long-range Interaction energy calculations

Starting from operators

To derive working formulas for interaction energy calculations one has to specify first the Hamiltonian $\mathscr{H}_{\text{int}}$ operator of interaction between charge distributions of interest. In principle, these interactions are *electrostatic* between all particles in the system, i.e. electrons and protons from nuclei. Since the interaction comes from electrostatics the exact Hamiltonian is pairwise additive and one can concentrate on the interaction between two molecules, A and B. It is important to remark here that this section is devoted to treating long-range intermolecular interactions (i.e. without taking into account overlap and exchange-repulsion effects explicitly). The resulting interaction Hamiltonian can be drawn as follows:

$$\mathscr{H}_{\text{int}} = \mathscr{H}_{\text{int}}^{\text{nuc}} + \mathscr{H}_{\text{int}}^{\text{el}-\text{nuc}} + \mathscr{H}_{\text{int}}^{\text{el}} \,, \tag{1}$$

where:

$$\begin{aligned}
\mathscr{H}_{\text{int}}^{\text{nuc}} &= \sum_{\alpha\in\text{A}}\sum_{\beta\in\text{B}}\frac{Z_\alpha Z_\beta}{r_{\alpha\beta}} && (2)\\
\mathscr{H}_{\text{int}}^{\text{el-nuc}} &= -\sum_{\alpha\in\text{A}}\sum_{j\in\text{B}}\frac{Z_\alpha}{r_{\alpha j}} - \sum_{\beta\in\text{B}}\sum_{i\in\text{A}}\frac{Z_\beta}{r_{\beta i}} && (3)\\
\mathscr{H}_{\text{int}}^{\text{el}} &= \sum_{i\in\text{A}}\sum_{j\in\text{B}}\frac{1}{r_{ij}} && (4)
\end{aligned}$$

In the above notation greek subscripts denote nuclei whereas latin subscripts distinguish electrons. An alternative but very useful form of $\mathscr{H}_{\text{int}}$ can be achieved using a charge density operator of molecule A, $\hat{\rho}^{\text{A}}$ which is defined as:

$$\hat{\rho}^{\text{A}}(\mathbf{r}) = \sum_{a\in\text{A}} e_a\delta(\mathbf{r}-\mathbf{a}) \;, \tag{5}$$

where the summation extends over all the particles (nuclei and electrons). Using 5 and integrating out nuclear charge densities[3] the respective interaction Hamiltonian terms can be written as follows:

$$\begin{aligned}
\mathscr{H}_{\text{int}}^{\text{nuc}} &= \sum_{\alpha\in\text{A}}\sum_{\beta\in\text{B}}\frac{Z_\alpha Z_\beta}{r_{\alpha\beta}} && (6)\\
\mathscr{H}_{\text{int}}^{\text{el-nuc}} &= -\sum_{\alpha\in\text{A}} Z_\alpha\int\frac{\hat{\rho}^{\text{B}}(\mathbf{r}_j)}{r_{\alpha j}}\text{d}\mathbf{r}_j - \sum_{\beta\in\text{B}} Z_\beta\int\frac{\hat{\rho}^{\text{A}}(\mathbf{r}_i)}{r_{\beta i}}\text{d}\mathbf{r}_i && (7)\\
\mathscr{H}_{\text{int}}^{\text{el}} &= \iint\frac{\hat{\rho}^{\text{A}}(\mathbf{r}_i)\hat{\rho}^{\text{B}}(\mathbf{r}_j)}{r_{ij}}\text{d}\mathbf{r}_j\text{d}\mathbf{r}_i \quad (i\in A \text{ and } j\in B) \;, && (8)
\end{aligned}$$

where $\hat{\rho}^{\text{A}}(\mathbf{r}_i)$ are in this case only electron charge density operators.

There is another approach, also sometimes very useful, based on multipole expansion. The interaction Hamiltonian can be written in the form:

$$\mathscr{H}_{\text{int}} = \sum_{a\in\text{A}} e_a V_{\text{B}}(\mathbf{a}) = \sum_{b\in\text{B}} e_b V_{\text{A}}(\mathbf{b}) \;, \tag{9}$$

where $V_{\text{X}}(\mathbf{r})$ is a electrostatic potential in point $\mathbf{r}$ generated by molecule X. Expanding this potential in Taylor series and applying traceless multipole moments the interaction Hamiltonian has the form (up to hexadecapole moments):

$$\mathscr{H}_{\text{int}} = \hat{q}^B\hat{V}^A + \hat{\mu}_a^B\hat{V}_a^A + \frac{1}{3}\hat{\Theta}_{ab}^B\hat{V}_{ab}^A + \frac{1}{15}\hat{\Omega}_{abc}^B\hat{V}_{abc}^A + \frac{1}{105}\hat{\Xi}_{abcd}^B\hat{V}_{abcd}^A \;, \tag{10}$$

where potential is expressed by a series of T-tensors (interaction tensors):

$$\hat{V}^X = \hat{q}^X\hat{T} + \hat{\mu}_a^X\hat{T}_a + \frac{1}{3}\hat{\Theta}_{ab}^X\hat{T}_{ab} + \frac{1}{15}\hat{\Omega}_{abc}^X\hat{T}_{abc} + \frac{1}{105}\hat{\Xi}_{abcd}^B\hat{T}_{abcd} \tag{11}$$

and

$$\begin{aligned}
\hat{T} &= \frac{1}{R} && (12)\\
\hat{T}_{ab\cdots x} &= \nabla_a\nabla_b\cdots\nabla_x\Big(\frac{1}{R}\Big) && (13)
\end{aligned}$$

[3]we use Born-Oppenheimer approximation

The potential gradient operators and higher-order derivatives $\hat{V}_{ab\cdots x}$ are related very simply with the potential operator $\hat{V}$:

$$\hat{V}_{ab\cdots x} = \nabla_a \nabla_b \cdots \nabla_x \hat{V} \tag{14}$$

Substituting into 10 the terms of potential and its derivatives and keeping in mind that we expand the interaction between A and B with respect to $\mathbf{R} = \mathbf{R}_\mathrm{B} - \mathbf{R}_\mathrm{A}$ (so we have to change the sign of the odd-rank $\hat{T}$-tensor operators for the potential in 11) we obtain[4]:

$$\begin{aligned}\mathscr{H}_\mathrm{int} &= \hat{q}^B\big[\hat{q}^A\hat{T} - \hat{\mu}_a^A\hat{T}_a + \frac{1}{3}\hat{\Theta}_{ab}^A\hat{T}_{ab} - \frac{1}{15}\hat{\Omega}_{abc}^A\hat{T}_{abc} + \frac{1}{105}\hat{\Xi}_{abcd}^A\hat{T}_{abcd}\big] \\ &+ \hat{\mu}_a^B\big[\hat{q}^A\hat{T}_a - \hat{\mu}_b^A\hat{T}_{ab} + \frac{1}{3}\hat{\Theta}_{bc}^A\hat{T}_{abc} - \frac{1}{15}\hat{\Omega}_{bcd}^A\hat{T}_{abcd}\big] \\ &+ \frac{1}{3}\hat{\Theta}_{ab}^B\big[\hat{q}^A\hat{T}_{ab} - \hat{\mu}_c^A\hat{T}_{abc} + \frac{1}{3}\hat{\Theta}_{cd}^A\hat{T}_{abcd}\big] \\ &+ \frac{1}{15}\hat{\Omega}_{abc}^B\big[\hat{q}^A\hat{T}_{abc} - \hat{\mu}_d^A\hat{T}_{abcd}\big] \\ &+ \frac{1}{105}\hat{\Xi}_{abcd}^B\big[\hat{q}^A\hat{T}_{abcd}\big]\end{aligned} \tag{15}$$

The above experesion can be rearranged into a more convenient form containing respective $\hat{T}$-tensor terms:

$$\begin{aligned}\mathscr{H}_\mathrm{int} &= \hat{T}\hat{q}^A\hat{q}^B \\ &+ \hat{T}_a\big(\hat{\mu}_a^B\hat{q}^A - \hat{q}^B\hat{\mu}_a^A\big) \\ &+ \hat{T}_{ab}\big(\frac{1}{3}\hat{q}^B\hat{\Theta}_{ab}^A + \frac{1}{3}\hat{q}^A\hat{\Theta}_{ab}^B - \hat{\mu}_b^A\hat{\mu}_a^B\big) \\ &+ \hat{T}_{abc}\big(\frac{1}{3}\hat{\mu}_a^B\hat{\Theta}_{bc}^A - \frac{1}{3}\hat{\Theta}_{ab}^B\hat{\mu}_c^A - \frac{1}{15}\hat{\Omega}_{abc}^A\hat{q}^B + \frac{1}{15}\hat{\Omega}_{abc}^B\hat{q}^A\big) \\ &+ \hat{T}_{abcd}\big(\frac{1}{9}\hat{\Theta}_{ab}^B\hat{\Theta}_{cd}^A - \frac{1}{15}\hat{\mu}_a^B\hat{\Omega}_{bcd}^A - \frac{1}{15}\hat{\Omega}_{abc}^B\hat{\mu}_d^A + \frac{1}{105}\hat{\Xi}_{abcd}^A\hat{q}^B + \frac{1}{105}\hat{q}^A\hat{\Xi}_{abcd}^B\big) \\ &= \mathscr{A}_0 + \mathscr{A}_1 + \mathscr{A}_2 + \mathscr{A}_3 + \mathscr{A}_4\end{aligned} \tag{16}$$

This Hamiltonian formulation will be used to obtain the working expression by rewriting interaction tensors explicitly in the subsection 3.3.

Perturbative treatment of intermolecular interactions

To calculate interaction energy one uses perturbation theory due to the fact that intermolecular interaction can be regarded as a perturbation of a total Hamiltonian for the two molecular systems, $\mathscr{H} = \mathscr{H}_\mathrm{A} + \mathscr{H}_\mathrm{B} + \mathscr{H}_\mathrm{int}$, where the first two terms accounts for unperturbed Hamiltonians of molecules A and B (as if they did not interact) and the last term defined as in previous paragraph describing the A–B interaction.

From the Raileygh-Schrödinger perturbation theory one can calculate the first-order interaction energy which equals:

$$E_\mathrm{int}^{(1)} = \big\langle \Psi_\mathrm{A}\Psi_\mathrm{B}\big|\mathscr{H}_\mathrm{int}\big|\Psi_\mathrm{A}\Psi_\mathrm{B}\big\rangle \tag{17}$$

To obtain interaction energy in the first order it has only to replace the operators from equations 23–28 by their respective expectation values between ground state functions In

[4]we take into account terms up to 4-th rank

order to calculate higher order corrections the expectation values of these tensor operators between functions including also excited states have to be considered. In COULOMB.PY only first-order interaction energies are available at present.

Using derived interaction Hamiltonian operator forms and applying perturbation theory to the first order one can obtain working formulae for the interaction energy (in the first-order) from Density Cube Method (IE DC), first-order electrostatics from Variational-perturbational Interaction Energy Decomposition Scheme (VP IEDS), Cumulative Atomic Multipole Moments Interaction Energy Decomposition Scheme (CAMM IEDS) and Interaction Energy from Electrostatic Potential Charge Fitting (IE ESP). Beneath, the respective derivations for above mentioned methods are listed.

3.1 Density Cube Method

Changing the expectation values of the electron density operators from 6 one can obtain the following expression for the first-order interaction energy:

$$E_{\mathrm{int}}^{(1)} = \sum_{\alpha\in\mathrm{A}}\sum_{\beta\in\mathrm{B}}\frac{Z_\alpha Z_\beta}{r_{\alpha\beta}} - \sum_{\alpha\in\mathrm{A}} Z_\alpha \int \frac{\rho^{\mathrm{B}}(\mathbf{r}_j)}{r_{\alpha j}}\mathrm{d}\mathbf{r}_j - \sum_{\beta\in\mathrm{B}} Z_\beta \int \frac{\rho^{\mathrm{A}}(\mathbf{r}_i)}{r_{\beta i}}\mathrm{d}\mathbf{r}_i + \iint \frac{\rho^{\mathrm{A}}(\mathbf{r}_i)\rho^{\mathrm{B}}(\mathbf{r}_j)}{r_{ij}}\mathrm{d}\mathbf{r}_j\mathrm{d}\mathbf{r}_i \tag{18}$$

This is the concept of Density Cube Method. In practice one uses `cube` files of electron density.

It has been shown that the method doesn't work at all when the density cubes overlap.

3.2 First-order electrostatics from VP IEDS

The same equations 6 can be rewritten in terms of electron density matrix obtained e.g. HF calculations and the optimized orbitals. Electron density in Restricted Hartee-Fock (RHF) scheme can be expressed by:

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}\phi_\mu^*(\mathbf{r})\phi_\nu(\mathbf{r}) \tag{19}$$

Substituting 19 into the integrals from 18 which consequently equal:

$$\begin{aligned} -Z_\iota \int \frac{\rho^X(\mathbf{r}_i)}{r_{\iota i}}\,\mathrm{d}\mathbf{r}_i &= -Z_\iota \int \Big[\sum_{\mu\nu} P_{\mu\nu}^X \phi_\mu^* \frac{1}{r_{\iota i}}\phi_\nu\Big]\,\mathrm{d}\mathbf{r}_i = \\ &= \sum_{\mu\nu} P_{\mu\nu}^X \Big[\int -Z_\iota \phi_\mu^* \frac{1}{r_{\iota i}}\phi_\nu\,\mathrm{d}\mathbf{r}_i\Big] = \sum_{\mu\nu} P_{\mu\nu}^X V_{\mu\nu}^\iota \end{aligned} \tag{20}$$

and

$$\begin{aligned} \iint \frac{\rho^X(\mathbf{r}_i)\rho^Y(\mathbf{r}_j)}{r_{ij}}\,\mathrm{d}\mathbf{r}_i\,\mathrm{d}\mathbf{r}_j &= \iint \Big[\sum_{\mu\nu} P_{\mu\nu}^X \phi_\mu^*\phi_\nu \frac{1}{r_{ij}} \sum_{\lambda\sigma} P_{\lambda\sigma}^Y \phi_\lambda^*\phi_\sigma\Big]\,\mathrm{d}\mathbf{r}_i\,\mathrm{d}\mathbf{r}_j = \\ &= \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}^X P_{\lambda\sigma}^Y \Big[\iint \phi_\mu^*\phi_\nu \frac{1}{r_{ij}}\phi_\lambda^*\phi_\sigma\,\mathrm{d}\mathbf{r}_i\,\mathrm{d}\mathbf{r}_j\Big] = \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}^X P_{\lambda\sigma}^Y (\mu\nu|\lambda\sigma) \end{aligned} \tag{21}$$

we arrive into the final $E_{\mathrm{el}}^{(1)}$ expression:

$$E_{\mathrm{el}}^{(1)} = \sum_{\alpha\in\mathrm{A}}\sum_{\beta\in\mathrm{B}}\frac{Z_\alpha Z_\beta}{r_{\alpha\beta}} + \sum_{\mu\nu\beta} P_{\mu\nu}^A V_{\mu\nu}^\beta + \sum_{\lambda\sigma\alpha} P_{\lambda\sigma}^B V_{\lambda\sigma}^\alpha + \sum_{\mu\nu\lambda\sigma} P_{\mu\nu}^A P_{\lambda\sigma}^B (\mu\nu|\lambda\sigma) \tag{22}$$

The resulting equation use RHF density matrices for molecules A and B calculated in Dimer Centered Basis Set (DCBS) in order to prevent non-physical effect of Basis Set Superposition Error (BSSE). There are also used nuclear attraction integrals $V^{\beta}_{\mu\nu}$ of electron from one molecule and nuclei from other as well as two-electron integrals $(\mu\nu|\lambda\sigma)$. All the integrals are used in Atomic Orbital (AO) representation.

3.3 Multipole part of electrostatic energy from CAMM IEDS

Substituting multpipole operators by their respective expectation values in 16 and summing over all distributed multipole moments in molecules A and B one obtains:

$$\begin{aligned}
E^{(1)}_{\mathrm{int}} &= \sum_{\alpha\in \mathrm{A}}\sum_{\beta\in \mathrm{B}}\Big[Tq^{\alpha}q^{\beta} && (23)\\
&+ T_a\big(\mu^{\beta}_a q^{\alpha} - q^{\beta}\mu^{\alpha}_a\big)\\
&+ T_{ab}\big(\frac{1}{3}q^{\beta}\Theta^{\alpha}_{ab} + \frac{1}{3}q^{\alpha}\Theta^{\beta}_{ab} - \mu^{\alpha}_b\mu^{\beta}_a\big)\\
&+ T_{abc}\big(\frac{1}{3}\mu^{\beta}_a\Theta^{\alpha}_{bc} - \frac{1}{3}\Theta^{\beta}_{ab}\mu^{\alpha}_c - \frac{1}{15}\Omega^{\alpha}_{abc}q^{\beta} + \frac{1}{15}\Omega^{\beta}_{abc}q^{\alpha}\big)\\
&+ T_{abcd}\big(\frac{1}{9}\Theta^{\beta}_{ab}\Theta^{\alpha}_{cd} - \frac{1}{15}\mu^{\beta}_a\Omega^{\alpha}_{bcd} - \frac{1}{15}\Omega^{\beta}_{abc}\mu^{\alpha}_d + \frac{1}{105}\Xi^{\alpha}_{abcd}q^{\beta} + \frac{1}{105}q^{\alpha}\Xi^{\beta}_{abcd}\big)\Big]\\
&= \sum_{\alpha\in \mathrm{A}}\sum_{\beta\in \mathrm{B}}\Big[A^{\alpha\beta}_0 + A^{\alpha\beta}_1 + A^{\alpha\beta}_2 + A^{\alpha\beta}_3 + A^{\alpha\beta}_4\Big]
\end{aligned}$$

The terms from $A^{\alpha\beta}_0$ to $A^{\alpha\beta}_4$ have to be calculated further by using interaction tensors in their explicit forms:

$$\begin{aligned}
T &= \frac{1}{R} && (24)\\
T_a &= \nabla_a\frac{1}{R} = -\frac{R_a}{R^3} && (25)\\
T_{ab} &= \nabla_a\nabla_b\frac{1}{R} = \frac{3R_aRb - R^2\delta_{ab}}{R^5} && (26)\\
T_{abc} &= \nabla_a\nabla_b\nabla_c\frac{1}{R} = \frac{15R_aR_bR_c - 3R^2(R_a\delta_{bc} + R_b\delta_{ac} + R_c\delta_{ab})}{R^7} && (27)\\
T_{abcd} &= \nabla_a\nabla_b\nabla_c\nabla_d\frac{1}{R} = && (28)\\
&= \frac{1}{R^9}\Big[105R_aR_bR_cR_d\\
&\quad -15R^2(R_aR_b\delta_{cd} + R_aR_c\delta_{bd} + R_aR_d\delta_{bc} + R_bR_c\delta_{ad} + R_bR_d\delta_{ac} + R_cR_d\delta_{ab})\\
&\quad +3R^4(\delta_{ab}\delta_{cd} + \delta_{ac}\delta_{bd} + \delta_{ad}\delta_{bc})\Big]
\end{aligned}$$

$A^{\alpha\beta}_0$ term

This term describes the interaction between two monopoles (charges) and it's very easy:

$$A^{\alpha\beta}_0 = \frac{q^{\alpha}q^{\beta}}{R}\ , \qquad (29)$$

where $R = \big|\mathbf{R}_{\beta} - \mathbf{R}_{\alpha}\big|$.

$A_1^{\alpha\beta}$ term

This term describes the interaction between monopole and dipole and equals:

$$A_1^{\alpha\beta} = -\frac{R_a}{R^3}\left[\mu_a^B q^A - q^B \mu_a^A\right] = \frac{q^\beta \boldsymbol{\mu}^\alpha \mathbf{R} - q^\alpha \boldsymbol{\mu}^\beta \mathbf{R}}{R^3} \tag{30}$$

$A_2^{\alpha\beta}$ term

This term describes the interactions between monopole and quadrupole as well as dipole-dipole interactions:

$$\begin{aligned} A_2^{\alpha\beta} &= \frac{3R_aR_b - R^2\delta_{ab}}{R^5}\left[\frac{1}{3}q^\beta\Theta^\alpha_{ab} + \frac{1}{3}q^\alpha\Theta^\beta_{ab} - \mu^\alpha_b\mu^\beta_a\right] = \frac{\boldsymbol{\mu}^\alpha\boldsymbol{\mu}^\beta}{R^3} - \\ &- 3\frac{(\mathbf{R}\boldsymbol{\mu}^\alpha)(\mathbf{R}\boldsymbol{\mu}^\beta)}{R^5} + \frac{q^\beta(\mathbf{R}\otimes_1 \boldsymbol{\Theta}^\alpha{}_2\otimes\mathbf{R}) + q^\alpha(\mathbf{R}\otimes_1\boldsymbol{\Theta}^\beta{}_2\otimes\mathbf{R})}{R^5} \end{aligned} \tag{31}$$

$A_3^{\alpha\beta}$ term

This term describes the interactions between dipole and quadrupole as well as monopole-octupole interactions and equals:

$$\begin{aligned} A_3^{\alpha\beta} &= \frac{15R_aR_bR_c - 3R^2(R_a\delta_{bc} + R_b\delta_{ac} + R_c\delta_{ab})}{R^7} \times \\ &\quad \times\left[\frac{1}{3}\mu^\beta_a\Theta^\alpha_{bc} - \frac{1}{3}\Theta^\beta_{ab}\mu^\alpha_c - \frac{1}{15}\Omega^\alpha_{abc}q^\beta + \frac{1}{15}\Omega^\beta_{abc}q^\alpha\right] = \\ &= 5\frac{R_c\mu^\alpha_c\cdot R_a\Theta^\beta_{ab}R_b - R_a\mu^\beta_a\cdot R_b\Theta^\alpha_{bc}R_c}{R^7} + \frac{q^\beta R_aR_bR_c\Omega^\alpha_{abc} - q^\alpha R_aR_bR_c\Omega^\beta_{abc}}{R^7} + \\ &\quad + \frac{\mu^\beta_a R_a\Theta^\alpha_{bb} + R_b\Theta^\alpha_{bc}\mu^\beta_c + \mu^\beta_b\Theta^\alpha_{bc}R_c}{R^5} - \frac{R_a\Theta^\beta_{ac}\mu^\alpha_c + \mu^\alpha_c\Theta^\beta_{cb}R_b + \mu^\alpha_c R_c\Theta^\beta_{bb}}{R^5} - \\ &\quad - \frac{q^\beta(R_a\Omega^\alpha_{acc} + R_b\Omega^\alpha_{cbc} + R_c\Omega^\alpha_{bbc})}{R^5} + \frac{q^\alpha(R_a\Omega^\beta_{acc} + R_b\Omega^\beta_{cbc} + R_c\Omega^\beta_{bbc})}{R^5} = \\ &= 5\frac{\mathbf{R}\boldsymbol{\mu}^\alpha(\mathbf{R}\otimes_1\boldsymbol{\Theta}^\beta{}_2\otimes\mathbf{R}) - \mathbf{R}\boldsymbol{\mu}^\beta(\mathbf{R}\otimes_1\boldsymbol{\Theta}^\alpha{}_2\otimes\mathbf{R})}{R^7} + \\ &\quad + \frac{q^\beta(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\boldsymbol{\Omega}^\alpha{}_3\otimes\mathbf{R}) - q^\alpha(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\boldsymbol{\Omega}^\beta{}_3\otimes\mathbf{R})}{R^7} + \\ &\quad + 2\frac{\boldsymbol{\mu}^\beta\otimes_1\boldsymbol{\Theta}^\alpha{}_2\otimes\mathbf{R} - \boldsymbol{\mu}^\alpha\otimes_1\boldsymbol{\Theta}^\beta{}_2\otimes\mathbf{R}}{R^5} \end{aligned} \tag{32}$$

In the following derivations we used the traceless property of quadrupole and octupole moments, i.e. $\Theta_{aa} = 0$ and $\Omega_{aab} = \Omega_{aba} = \Omega_{baa} = 0$ for all combinations of subscripts.

$A_4^{\alpha\beta}$ term

This term describes the interactions between dipole and octupole, quadrupole-quadrupole as well as monopole and hexadecapole interactions:

$$
\begin{aligned}
A_4^{\alpha\beta} \; = \; & \frac{1}{R^9}\Big[105 R_a R_b R_c R_d \\
& -15R^2(R_aR_b\delta_{cd} + R_aR_c\delta_{bd} + R_aR_d\delta_{bc} + R_bR_c\delta_{ad} + R_bR_d\delta_{ac} + R_cR_d\delta_{ab}) \\
& +3R^4(\delta_{ab}\delta_{cd} + \delta_{ac}\delta_{bd} + \delta_{ad}\delta_{bc})\Big] \times \\
& \quad \times\Big[\frac{1}{9}\Theta^\beta_{ab}\Theta^\alpha_{cd} - \frac{1}{15}\mu^\beta_a\Omega^\alpha_{bcd} - \frac{1}{15}\Omega^\beta_{abc}\mu^\alpha_d + \frac{1}{105}\Xi^\alpha_{abcd}q^\beta + \frac{1}{105}q^\alpha\Xi^\beta_{abcd}\Big] = \\
= \; & \frac{35}{3}\frac{R_a\Theta^\alpha_{ab}R_b \cdot R_c\Theta^\beta_{cd}R_d}{R^9} - 7\frac{\mu^\beta_a R_a \cdot R_bR_cR_d\Omega^\alpha_{bcd} + R_aR_bR_c\Omega^\beta_{abc}\cdot\mu^\alpha_d R_d}{R^9} - \\
& \quad -\frac{15}{9}\frac{R_a\Theta^\beta_{ad}\Theta^\alpha_{dc}R_c + R_b\Theta^\beta_{bd}\Theta^\alpha_{dc}R_c + R_b\Theta^\beta_{ba}\Theta^\alpha_{ad}R_d + R_a\Theta^\beta_{ac}\Theta^\alpha_{cd}R_d}{R^7} + \\
& \quad +\frac{R_bR_c\Omega^\alpha_{bca}\mu^\beta_a + R_cR_d\Omega^\alpha_{cda}\mu^\beta_a + R_bR_d\Omega^\alpha_{bda}\mu^\beta_a}{R^7} + \\
& \quad +\frac{R_bR_c\Omega^\beta_{bca}\mu^\alpha_a + R_cR_d\Omega^\beta_{cda}\mu^\alpha_a + R_bR_d\Omega^\beta_{bda}\mu^\alpha_a}{R^7} + \\
& \quad +\frac{1}{3}\frac{\Theta^\beta_{cd}\Theta^\alpha_{cd} + \Theta^\beta_{cd}\Theta^\alpha_{cd}}{R^5} + \frac{R_aR_bR_cR_d(q^\alpha\Xi^\beta_{abcd} + \Xi^\alpha_{abcd}q^\beta)}{R^9} + \\
& \quad + \text{remaining zero terms with traces of } \Omega \text{ and } \Xi = \\
= \; & \frac{35}{3}\frac{(\mathbf{R}\otimes_1 \boldsymbol{\Theta}^\alpha{}_2 \otimes \mathbf{R})\cdot(\mathbf{R}\otimes_1 \boldsymbol{\Theta}^\beta{}_2 \otimes \mathbf{R})}{R^9} - \\
& \quad -7\frac{(\boldsymbol{\mu}^\beta\mathbf{R})\cdot(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\boldsymbol{\Omega}^\alpha{}_3\otimes\mathbf{R}) + (\boldsymbol{\mu}^\alpha\mathbf{R})\cdot(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\boldsymbol{\Omega}^\beta{}_3\otimes\mathbf{R})}{R^9} - \\
& \quad -\frac{20}{3}\frac{(\mathbf{R}\otimes_1\boldsymbol{\Theta}^\alpha)\cdot(\mathbf{R}\otimes_1\boldsymbol{\Theta}^\beta)}{R^7} + \frac{2}{3}\frac{(\boldsymbol{\Theta}^\alpha{}_{12}\otimes_{12}\boldsymbol{\Theta}^\beta)}{R^5} + \\
& \quad +3\frac{(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\boldsymbol{\Omega}^\alpha{}_3\otimes\boldsymbol{\mu}^\beta) + (\mathbf{R}\otimes_1\mathbf{R}\otimes_2\boldsymbol{\Omega}^\beta{}_3\otimes\boldsymbol{\mu}^\alpha)}{R^7} + \\
& \quad +\frac{q^\beta(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\mathbf{R}\otimes_3\boldsymbol{\Xi}^\alpha{}_4\otimes\mathbf{R}) + q^\alpha(\mathbf{R}\otimes_1\mathbf{R}\otimes_2\mathbf{R}\otimes_3\boldsymbol{\Xi}^\beta{}_4\otimes\mathbf{R})}{R^9}
\end{aligned} \tag{33}
$$

These working formulae are implemented in COULOMB.PY standard routines except for the terms containing hexadecapole moments.

3.4 Electrostatic energy from ESP

Having the charges fitted by applying ESP procedure the first-order interaction energy simply equals:

$$
E^{(1)}_{\text{int}} = \sum_{\alpha\in A}\sum_{\beta\in B}\frac{q^\alpha q^\beta}{r_{\alpha\beta}} \tag{34}
$$

3.5 DFI pseudo-coulomb interaction energy

DFI method differs strongly from the previous ones primarily because it involves not only classical electrostatic interaction but also intermolecular first-order exchange-repulsion (from exchange integrals between electrons of respective molecules or, here, *fragments*) as well as first-order induction and delocalisation by applying self-consistent procedure for optimisation DFI Hamiltonian matrix. The next very important benefit is that it allows to study the interactions between *more than two* molecules. Therefore it enables to take into account many-body effects. Due to the fact that previous methods are designed to calculate only coulomb iteraction energy, the DFI interaction energy is called *pseudo-coulomb*.

4 Auxiliary method overview

4.1 Cumulative Atomic Multipole Moments

Cumulative Atomic Multipole Moments (CAMMs) are cartesian atomic multipole moments distributed and centered at the atomic positions and can be expressed recursively by a change of coordinate system origin from the molecular center to the respective atomic centers:

$$M^{\mathrm{CAMM}}_{klm,i} = M_{klm,i} - \sum_{k'\geq 0}^{k\neq k'} \sum_{l'\geq 0}^{l\neq l'} \sum_{m'\geq 0}^{m\neq m'} \binom{k}{k'}\binom{l}{l'}\binom{m}{m'} \times x_i^{k-k'} y_i^{l-l'} z_i^{m-m'} M^{\mathrm{CAMM}}_{k'l'm',i} \tag{35}$$

The transformed atomic multipole moments $M_{klm,i}$ all centered at molecular origin and distributed at i-th atom are described as:

$$M_{klm,i} = Z_i x_i^k y_i^l z_i^m - \sum_{\mu\in i}\sum_{\nu} P_{\mu\nu}\langle\mu|x^k y^l z^m|\nu\rangle \tag{36}$$

After applying equations 35 and 36 for evaluation of the first four cumulative atomic multipoles one obtains the following working formulae:

$$q^i = Z_i - \sum_{\mu\in i}\sum_{\nu} P_{\mu\nu}S_{\mu\nu} \tag{37}$$

$$\mu_a^i = \sum_{\mu\in i}\sum_{\nu} P_{\mu\nu}\left[a_i S_{\mu\nu} - \langle\mu|a|\nu\rangle\right] \tag{38}$$

$$Q_{ab}^i = \sum_{\mu\in i}\sum_{\nu} P_{\mu\nu}\left[-a_i b_i S_{\mu\nu} + b_i\langle\mu|a|\nu\rangle + a_i\langle\mu|b|\nu\rangle - \langle\mu|ab|\nu\rangle\right] \tag{39}$$

$$\begin{aligned}\Omega_{abc}^i = \sum_{\mu\in i}\sum_{\nu} P_{\mu\nu}&\left[a_i b_i c_i S_{\mu\nu} - b_i c_i\langle\mu|a|\nu\rangle - a_i c_i\langle\mu|b|\nu\rangle - a_i b_i\langle\mu|c|\nu\rangle\right.\\ &\left.+c_i\langle\mu|ab|\nu\rangle + b_i\langle\mu|ac|\nu\rangle + a_i\langle\mu|bc|\nu\rangle - \langle\mu|abc|\nu\rangle\right]\end{aligned} \tag{40}$$

Traceless form of CAMMs

Multpole n-rank tensor operators in 'ordinary' cartesian form are defined as follows:

$$\mathbf{m}^{(n)} = \sum_i e_i \underbrace{\mathbf{r}_i \otimes \cdots \otimes \mathbf{r}_i}_{n} \tag{41}$$

For instance, dipole, quadrupole and octupole tensor operator elements are:

$$\mu_a = \sum_i e_i a_i \tag{42}$$

$$Q_{ab} = \sum_i e_i a_i b_i \tag{43}$$

$$\Omega_{abc} = \sum_i e_i a_i b_i c_i \quad \text{where} \quad a,b,c = x,y \text{ or } z \tag{44}$$

In many applications such a form of definition in the case of quadrupole and higher multipole moments leads to complicated formulas wich are difficult to operate in practice. This problem can be solved by using so called *traceless form* of these operators which make them convenient in use. For quadrupole moment operator traceless forms is defined as follows:

$$Q'_{ab} = \frac{1}{2}\sum_i e_i\big(3a_ib_i - r_i^2\delta_{ab}\big) \tag{45}$$

In the case of octupole moment the definition is a bit more complex:

$$\Omega'_{abc} = \frac{1}{2}\sum_i e_i\big(5a_ib_ic_i - r_i^2(a_i\delta_{bc} + b_i\delta_{ac} + c_i\delta_{ab})\big) \tag{46}$$

In the above equations r_i^2 equals:

$$r_i^2 = \sum_{a=\{x,y,z\}} {a_i}^2 \tag{47}$$

It is very easy to switch between 'ordinary' tensor form to traceless one and *vice versa.* The transition to traceless quadrupole moment from ordinary cartesian quadrupole moment can be done like in the example below. Rewriting the equation 45 and substituting the definition from 42 and 47 one quickly obtains:

$$\begin{aligned} Q'_{ab} &= \frac{3}{2}\sum_i e_ia_ib_i - \frac{1}{2}\sum_i\sum_\alpha e_i\alpha_i^2\delta_{ab} = \frac{3}{2}Q_{ab} - \delta_{ab}\sum_\alpha\sum_i e_i\alpha_i^2 = \\ &= \frac{3}{2}Q_{ab} - \delta_{ab}\ \mathrm{Tr}\ Q_{ab} \end{aligned} \tag{48}$$

Similar calculations for traceless octupole moment give following expression:

$$\begin{aligned} \Omega'_{abc} &= \frac{5}{2}\sum_i e_ia_ib_ic_i - \frac{1}{2}\sum_i\sum_\alpha e_i\alpha_i^2\big(a_i\delta_{bc} + b_i\delta_{ac} + c_i\delta_{ab}\big) = \\ &= \frac{5}{2}\Omega_{abc} - \frac{1}{2}\big(\Omega_{\alpha\alpha a}\delta_{bc} + \Omega_{\alpha\alpha b}\delta_{ac} + \Omega_{\alpha\alpha c}\delta_{ab}\big) \end{aligned} \tag{49}$$

where $\Omega_{\alpha\alpha c}$ denotes trace of $\mathbf{\Omega}$ tensor with respect to the first two axes written in Einstein summation notation for simplicity. The monopole, dipole, quadrupole and octupole moments from equations 3, 4, 13 and 14 are used in standard COULOMB.PY routines for the calculations of electrostatic potential $V(\mathbf{r})$ as well as multipole part of electrostatic interaction energy $\Delta E_{el}^{\mathrm{MTP}}$ between two multipole distributions. For more details see the section 3 about methods of calculations of electrostatic potential and interaction energy.

4.2 Fitting charges from electrostatic potential

The ESP method is used for calculation of coulomb interaction between charge distributions or in assigning the partial charges on atoms in force field evaluation for molecular dynamics purposes. The key problem in ESP is to perform the best possible fitting of the charges to obtain the reference potential (in COULOMB.PY either from CAMMs or wave function).

The function $\mathscr{X} = \left|\mathbf{V}_{\mathrm{ref}} - \mathbf{A}\mathbf{q}^{\mathrm{T}}\right|$ has to be minimized, where $\mathbf{V}_{\mathrm{ref}}$ is reference potential, $\mathbf{q}$ is a vector of charges treated as variational parameters and $\mathbf{A}$ is the distance matrix between these charges, $A_{ij} = 1/r_{ij}$. The convenient way to find $\mathbf{q}$ leads through least-square procedure. The squares of the difference of calculated potential has to be minimized with a constraint of constant total charge:

$$Z(\{q_j\}) = \sum_i^m \Big(V_i - \sum_j^N \frac{q_j}{r_{ij}}\Big)^2 + \lambda\Big(\sum_j^N q_j - q_{tot}\Big) \tag{50}$$

There is m points in which electrostatic potential has to be estimated[5] and N charges q_j to be fit (usually on atoms). The constraint of the least-square fitting is here the total charge, q_{tot} with the associated undetermined Lagrange multiplier, λ.

Minimisation of the function $Z(\{q_j\})$ is equivalent to take the first derivatives of Z with respect to fitted charges q_k and to make it be equal to zero:

$$\left(\frac{\partial Z(\{q_j\})}{\partial q_k}\right)_{q_{(i\neq k)}} = 0 \tag{51}$$

Thus we have:

$$\frac{\partial}{\partial q_k}\left[\sum_i^m \Big(V_i - \sum_j^N \frac{q_j}{r_{ij}}\Big)^2 + \lambda\Big(\sum_j^N q_j - q_{tot}\Big)\right] = 2\sum_i^m \Big(V_i - \sum_j^N \frac{q_j}{r_{ij}}\Big)\Big(-\frac{1}{r_{ik}}\Big) + \lambda = \tag{52}$$

$$\sum_j^N\sum_i^m \frac{q_j}{r_{ij}r_{ik}} - \sum_i^m \frac{V_i}{r_{ik}} + \lambda = 0$$

This equation can be written in a matrix form as follows:

$$\begin{pmatrix}\mathbf{A} & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\mathbf{q}\\ \lambda\end{pmatrix} = \begin{pmatrix}\mathbf{B}\\ 0\end{pmatrix}, \tag{53}$$

where the matrix elements are as follows:

$$A_{jk} = \sum_i^m \frac{1}{r_{ij}r_{ik}} \tag{54}$$

$$B_k = \sum_i^m \frac{V_i}{r_{ik}} \tag{55}$$

Inverting the matrix thad multiplies the vector with charges and λ we get the fitted charges as a result.

4.3 Density Fragment Interaction Method

The DFI method is based on Hartree-Fock-Roothaan-Hall equasion, when Fock matrix is diagonalized iteratively to obtain self-consistent density matrix and other useful variables. The DFI scheme extends Fock matrix by taking into account the interactions between some pieces of an entire system called *fragments* as well as bath-system interactions. The entire Fock matrix comprises of fragment block Fock matrices. Thus, the modified Fock matrix for a given fragment A can be written as follows:

$$\mathbf{F}_A^{\mathrm{DFI}} = \mathbf{F}_A + \sum_{B\neq A}\mathbf{V}_{AB} + \mathbf{V}_{A-\mathrm{bath}} \tag{56}$$

[5]in COULOMB.PY either from CAMMs or directly from wave function. See section 3.

where $\mathbf{F}_A$ stands for gas-phase 'ordinary' Fock matrix, $\mathbf{V}_{AB}$ is the A-B-interfragment interaction and $\mathbf{V}_{A-\mathrm{bath}}$ denotes A-bath coupling. The explicit formulae for the respective interactions are:

$$\mathbf{V}_{AB} = \boldsymbol{\mu}_{AB} + \boldsymbol{\nu}_{AB} \quad \text{where} \tag{57}$$

$$\boldsymbol{\mu}_{AB} = -\sum_{b\in B} \mathbf{V}_A^b \tag{58}$$

$$\boldsymbol{\nu}_{AB} = \mathbf{P_B}(\mathbf{J}(BA) - \frac{1}{2}\mathbf{K}(BA)) \tag{59}$$

$$J(BA)^{\mu\nu\in A}_{\lambda\sigma\in B} = (\mu\nu|\lambda\sigma) \tag{60}$$

$$K(BA)^{\mu\nu\in A}_{\lambda\sigma\in B} = (\mu\sigma|\lambda\nu) \tag{61}$$

DFI scheme adopts self consistent procedure to consider equally each of taken fragments. It is performed as follows:

1. $\mathbf{F}_A^{\mathrm{DFI}}$ for fragment A is calculated from separate gas-phase density matrices $\mathbf{P}_i$ of fragments i. $\mathbf{P}_A$ density matrix is temporally set to zero in order to avoid self-coulombic interactions of this fragment.

2. SCF procedure is applied on $\mathbf{F}_A^{\mathrm{DFI}}$ and new, DFI density matrix for fragment A is obtained, $\mathbf{P}_A^{\mathrm{DFI}}$

3. We switch to the next fragment, B. We construct for it $\mathbf{F}_B^{\mathrm{DFI}}$ from $\mathbf{P}_A^{\mathrm{DFI}}$, $\mathbf{P}_B$ taken as zero and other $\mathbf{P}_i$.

4. SCF procedure is applied on $\mathbf{F}_B^{\mathrm{DFI}}$ and now new, DFI density matrix for fragment B is obtained, $\mathbf{P}_B^{\mathrm{DFI}}$

5. We iterate like this until each of the fragments has been considered.

After this procedure the respective DFI density matrices $\mathbf{P}_i^{\mathrm{DFI}}$ as well as interfragment interaction potentials $\mathbf{V}_{ij}$ are obtained.