FDB.py manual

An efficient methodology to compute field-dependent chemical barriers

Version 1.0

Pau Besalú Sala, PhD candidate

Institut de Química Computacional i Catàlisi and
Universitat de Girona, Departament de Química,
Girona, Catalunya, Spain.
Contact through: paubesalu@gmail.com

Table of Contents

Chapter 1

First steps

1.1 Download and installation

The python script FDB.py can be downloaded free of charge at:

In order to execute the script, currently the following dependencies are required:

1. python 3.8.3 or superior
2. numpy 1.19.2 or superior

1.2 Required input files

There are three main input files that must be provided in the working directory.

1. fdb.inp
2. reactants.dat
3. ts.dat

In the fdb.inp file there is the neccessary information for the fdb.py script to work. It contains a variable number of lines, each one with a number (integer or float) which indicates what the program must do. The first block of instructions correspond to a single-axis calculation, and the second corresponds to a second-axis calculation.

In the reactants.dat file it must be provided the information about the energy and the derivatives of the energy respect to the electric field (EF) for the reactant complex geometry.

In the ts.dat file it must be provided the information about the energy and the derivatives of the energy respect to the electric field (EF) for the transition state geometry.

Disclaimer: The tensors for the dipole moment (μ), the polarizability (α) and the 1^{st} hyperpolarizability (β) are considered symmetric. This is the case when the input values are taken form a Gaussian16 calculation, and only the lower-triangular part of the matrix/tensor is required to be input.

All results will be displayed on screen by typing in the command line ”python fdb.py” or else echoed in a file.out by typing in the command line ”python fdb.py > file.out”

1.3 reactants and ts data

While the construction of the input file (fdb.inp) is discuted in detail avobe, the information that must be provided in the reactants.dat and ts.dat files is always the same, regardless of the calculation mode, whether is single- or double-axis calculation. Therefore, the files reactants.dat and ts.dat are constructed as follows:

In the first line, there must be the (electronic) energy of the chemical species (either the reactant complex or the transitio state) in atomic units. Type = float.

From second to fourth line, there must be given the dipole moment of the chemical species (in atomic units). In this block, the first line corresponds to μ_x, the second to μ_y and the third to μ_z. Type = float.

This is the minimum input required (Only dipole approximation). However, depending on which approximation the user requests on fdb.inp, the appropriate data should be also input.

In case the working approximation is dipole+polarizability(el) approximation (see below), the electronic polarizability tensor must be provided after the previous information. Specifically, this would imply that in the reactants.dat and ts.dat files, the lines 5 to 10 correspond to the following elements of the polarizability matrix (in atomic units, Type = float):

1. α_{xx}
2. $\alpha_{yx} (= \alpha_{xy})$
3. α_{yy}
4. $\alpha_{zx} (= \alpha_{xz})$
5. $\alpha_{zy} (= \alpha_{zy})$
6. α_{zz}

Please, notice that since the code assumes that the matrix is symmetric, there is only need to input 6 out of 9 elements of the matrix.

In case the working approximation is dipole+polarizability(el)+1^{st} hyperpolarizability approximation (see below), the electronic 1^{st} (hyper)polarizability tensor must be provided after the previous information. Specifically, this would imply that in the reactants.dat and ts.dat files, the lines 11 to 20 correspond to the following elements of the polarizability tensor (in atomic units, Type = float):

1. β_{xxx}
2. $\beta_{xxy} (= \beta_{xyx} = \beta_{yxx})$
3. $\beta_{yxy} (= \beta_{yyx} = \beta_{xyy})$
4. β_{yyy}
5. $\beta_{xxz} (= \beta_{xzx} = \beta_{zxx})$
6. $\beta_{yxz} (= \beta_{yzx} = \beta_{xyz} = \beta_{xzy} = \beta_{zyx} = \beta_{zxy})$
7. $\beta_{yyz} (= \beta_{yzy} = \beta_{zyy})$
8. $\beta_{zxz} (= \beta_{zzx} = \beta_{xzz})$
9. $\beta_{zyz} (= \beta_{zzy} = \beta_{yzz})$
10. β_{zzz}

Please, notice that since the code assumes that the tensor is symmetric, there is only need to input 10 out of 27 elements of the tensor. Furthermore, Gaussian16 code has an internal ill-definition of the sign for the whole tensor. The property is changed the sign automatically by the program, so if your input is computed by any other electronic structure package, please introduce the input multiplyed by (-1).

In case the working approximation is dipole+polarizability(el+nr)+1^{st} hyperpolarizability approximation (see below), the nuclear relaxation contribution to the polarizability matrix must be provided. In this case, right after the input of the electronic polarizability in the same format as the usual electronic polarizability is introduced (this means, from lines 11 to 16). this will shift the introduction of the β elements to lines 17 to 26. In atomic units, Type = float:

Chapter 2

Single-axis calculation

2.1 Working Mode (WM)

The working mode is the approximation that will be used to compute the field-dependent energy barrier.

The Taylor expansion (up to the 3rd term) of the chemical barrier is the following:

For the single axis calculations the user must select which approximation to use. The four available WM are:

1. Only dipole (WM=1, Eq. 2.1)

$$\Delta E(F) \cong \Delta E_0 - \sum_{i}^{x,y,z} \Delta\mu_i F_i \tag{2.1}$$

2. Dipole + electronic polarizability (WM=2, Eq. 2.2)

$$\Delta E(F) \cong \Delta E_0 - \sum_{i}^{x,y,z} \Delta\mu_i F_i - \frac{1}{2}\sum_{i,j}^{x,y,z} \Delta\alpha_{ij} F_i F_j \tag{2.2}$$

3. Dipole + electronic polarizability+1sthyperpolarizability (WM=3, Eq. 3.1)

$$\Delta E(F) \cong \Delta E_0 - \sum_{i}^{x,y,z} \Delta\mu_i F_i - \frac{1}{2}\sum_{i,j}^{x,y,z} \Delta\alpha_{ij} F_i F_j - \frac{1}{3!}\sum_{i,j,k}^{x,y,z} \Delta\beta_{ijk} F_i F_j F_k \tag{2.3}$$

4. Dipole + full (electronic+nuclear relaxation) polarizability+1sthyperpolarizability (WM=4, Eq. 3.1)

The difference between WM3 and WM4 is the way in which de polarizability matrix is constructed. In WM3 only the electronic part of the matrix is taken into account, whereas in WM4 also the nuclear relaxation contribution is summed into the polarizability matrix. This must be provided as two matrix (see above).

2.2 Run a single-axis calculation

The input file (fdb.inp) for a single-axis calculation run is computed in the following manner: The first line corresponds to the Working mode (which can be an integer ranging from 1 to 4), the second line corresponds to the number of electric fields explored (integer), the following lines correspond to the electric fields that want to be explored. The following (last) line corresponds to the WM for the double-axis calculation, if the user is not interested on that, just disable by input a zero. As an example, the fdb.inp for a WM=2 calculation, with 5 electric fields (-0.001, -0.0005, 0.0, 0.0005 and -0.001 a.u.) would be:

```
2         #Working mode
5         #Number of EF to explore (in all directions of 3D space)
-0.001    #First EF strength
-0.0005   #Second EF strength
0.0       #Third EF strength
0.0005    #Fourth EF strength
0.001     #Fifth EF strength
0         #Indicates that no double-axis calculation is desired
```

The display of the several barriers computed will be in the same order as the electric fields are provided.

Notice that for a single-axis run, the working equations simplify down to (eg. case, WM=3):

$$\Delta E(F) \cong \Delta E_0 - \Delta\mu_x F_x - \frac{1}{2}\Delta\alpha_{xx}F_x^2 - \frac{1}{3!}\Delta\beta_{xxx}F_x^3 \tag{2.4}$$

$$\Delta E(F) \cong \Delta E_0 - \Delta\mu_y F_y - \frac{1}{2}\Delta\alpha_{yy}F_y^2 - \frac{1}{3!}\Delta\beta_{yyy}F_y^3 \tag{2.5}$$

$$\Delta E(F) \cong \Delta E_0 - \Delta\mu_z F_z - \frac{1}{2}\Delta\alpha_{zz}F_z^2 - \frac{1}{3!}\Delta\beta_{zzz}F_z^3 \tag{2.6}$$

All three values for $\Delta E(F)$ are given in the same run.

Chapter 3

Double-axis calculation

For the double axis calculation, a single axis calculation must be done first to read the input data. Therefore, the first part of the input file must be the same as for a normal single-axis run. After that part, 5 more lines must be given, with the following information:

1. Working mode (WM). Type = integer ranging from 1 to 4, each number indicating the same working equation than in the single-axis mode. Disable the run of double-axis calculation by input a zero here (no further lines will be read).

2. axis to evaluate. Type = integer ranging from 1 to 3. 1=xy calculation, 2=xz calculation and 3=yz calculation.

3. First field to explore (the lowest field, in a.u.). Type = float

4. Stepsize between fields (density of the grid). Type = Float

5. Number of steps to take. Type = Integer

In this case, the working equations simplify down to (example for an xz run in WM=3):

$$\Delta E(F) \cong \Delta E_0 - \sum_{i}^{x,z} \Delta\mu_i F_i - \frac{1}{2}\sum_{i,j}^{x,z} \Delta\alpha_{ij} F_i F_j - \frac{1}{3!}\sum_{i,j,k}^{x,z} \Delta\beta_{ijk} F_i F_j F_k \qquad (3.1)$$

Notice that by playing with the three last values of the input file, all possible combinations for square grids are allowed.

An example of input for a single-axis (compulsory) + a double-axis run is given:

```
2         #Working mode. Single-axis run input
5         #Number of EF to explore (in all directions of 3D space).
-0.001    #First EF strength
-0.0005   #Second EF strength
0.0       #Third EF strength
0.0005    #Fourth EF strength
0.001     #Fifth EF strength
3         #Working mode. Here starts the double-axis run.
2         #Axis to compute, in this case xz.
-0.001    #First field to explore. Double-axis run.
0.001     #Stepsize for the fields. Double-axis run.
5         #Number of steps to take. Double-axis run.
```

In the final display of the results, if the requested number of steps to take is lower than 6, a matrix-like echoing of the results is given. This is useful for fast screenings. Once the region of interest is located, it can be explored further in another run with more points and more dense grid. In the case of having more than 6 points, the output will be displayed element by element in the following manner (ficticious example for an xz-calculation with 3 field-strengths):

$$\begin{gathered}
\Delta E(x1, z1) \\
\Delta E(x1, z2) \\
\Delta E(x1, z3) \\
\Delta E(x2, z1) \\
\Delta E(x2, z2) \\
\Delta E(x2, z3) \\
\Delta E(x3, z1) \\
\Delta E(x3, z2) \\
\Delta E(x3, z3)
\end{gathered}$$