JACoPO
Just Another COupling Program, Obviously

Daniele Padula[*1] and Marco Campetella[†1]

[1]*Dipartimento di Chimica e Chimica Industriale, Università di Pisa, via G. Moruzzi 13, 56124 Pisa, Italy*

19th May 2016

1   Presentation

`JACoPO` is a program to calculate coulombic electronic couplings between two excited states according to various approaches. In particular, it is possible to calculate couplings with the Transition Density Cube method, with the monopole–monopole approximation and with the dipole–dipole approximation (also known as Point Dipole Approximation, PDA).

2   Requirements

`JACoPO` is a program written in Python and Fortran 90. A working installation of python with numpy and a fortran compiler (such as gfortran) are the only requirements.

3   Installation

`JACoPO` should run on both Windows and Unix platforms. However, since the program has not been tested on a Windows machine, all the steps reported below imply that you are using a Unix machine.

To install `JACoPO` on your machine, few simple steps are needed. To get a copy of the program, there are a couple of alternatives. You could clone the git repository of the program:

```
$ git clone https://github.com/dpadula85/JACoPO
```

This way, you can always be updated to the last changes to the program by running
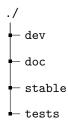
```
$ git pull
```

in the main directory of the repo.

Alternatively, you could browse to the address `https://github.com/dpadula85/JACoPO` and download a `.zip` file with the appropriate button.

The directory structure of the program is as follows

---

[*]dpadula85@yahoo.it
[†]marco.campetella82@gmail.com

```
./
├── dev
├── doc
├── stable
├── tests
```

The `dev` and `stable` directories should be the same, apart from experimental features under development in the `dev` folder. We recommend you use the executables in the `stable` folder. Make sure that the python files in this directory (`.py` extension) are executable. If they are not, run the commands

```
$ cd stable
$ chmod +x *.py
```

The program is distributed with a pre-compiled version of the needed Fortran 90 module, so you should be already able to run it. If you obtain some error related to this module or in case you prefer to recompile it, run the commands

```
$ make clean
$ make
```

in the `stable` directory. If everything worked fine, you can skip the rest of this section, since it deals with troubleshooting in this phase.

In some cases, we experienced problems with some numpy installations that were not respoding correctly to this command. If this is your case, try running

```
$ make clean
$ make alternative
```

If you keep getting errors, please contact us. Once these steps are done, `JACoPO` should be ready to work!

4 Running tests

In case you want to check whether everything is working properly, we suggest you run the tests distributed with the program. To do so, go in the `tests` directory and run the command

```
$ make &
```

The tests can take some time, but hopefully everything is working correctly! The result of the tests is summarized in the file `tests/logfile`, which is written on the fly while tests are running. You can check the progression of the tests in this file and which ones passed or failed by running

```
$ cat logfile
```

For more details about the tests, feel free to change directory to each test and have a look at the files, also for comparison with the instructions given in section 7.

5   Theory

Point Dipole Approximation (PDA)

This approximation is the one traditionally used to evaluate the coulombic term of electronic couplings. This description is based on a dipolar interaction.[1] Given two molecules D and A, the coulombic coupling between excited state a of molecule D and excited state b of molecule A is given by

$$V_{Da,Ab}^{coul} = \frac{1}{4\pi\varepsilon_0} \left(\frac{\vec{\mu_a} \cdot \vec{\mu_b}}{r_{DA}^3} - \frac{3(\vec{\mu_a} \cdot \vec{r}_{DA}) \cdot (\vec{\mu_b} \cdot \vec{r}_{DA})}{r_{DA}^5} \right) \tag{1}$$

where $\vec{\mu_a}$ is the transition electric dipole moment for excited state a of molecule D and r_{DA} is the distance between the two application points of the dipoles. Within this approximation, it is possible to define the orientation factor κ between the two chromophores as in

$$\kappa = \hat{\mu}_a \cdot \hat{\mu}_b - 3(\hat{\mu}_a \cdot \hat{r}_{DA}) \cdot (\hat{\mu}_b \cdot \hat{r}_{DA}) \tag{2}$$

This approximation falls off when the distance between the two molecules is comparable to the molecular size or when the molecule has an elongated structure, such as a carothenoid.

More refined approaches have been developed to deal with these limitations.

Monopole–Monopole Approximation

In this approximation, the Transition Density of an excited state is expressed in terms of point charges on the atoms constituting the molecule.[2] There are many methods to evaluate these point charges,[3] and the analysis of such methods goes beyond the scope of this manual. Once the set of charges is obtained with the method of choice, the coulombic term of the electronic coupling is calculated as

$$V_{Da,Ab}^{coul} = \frac{1}{4\pi\varepsilon_0} \sum_{i,j} \frac{q_D^a(i) \cdot q_A^b(j)}{r_{ij}} \tag{3}$$

where i runs over the coordinates of the atoms in molecule D and r_{ij} is the distance between atoms at positions i and j.

Within this approximation, the transition electric dipole moment associated to the charges can be obtained as

$$\vec{\mu_a} = \sum_i q^a(i)\vec{r_i} \tag{4}$$

thus the two transition electric dipole moments μ_a and μ_b can be retrieved by the two sets of charges and the coupling and the orientation factor can be estimated also according to the PDA.

Transition Density Cube method

This method was proposed in 1998 by Krueger $et\ al.$,[4] and resulted in the development of a program for the calculation of electronic coupling according to this method (see `http://www.chem.hope.edu/~krieg/TDC/TDC_home.htm`).

Briefly, this method evaluates the coulombic interaction between the transition densities of two molecules. The coulombic term of the coupling is given by

$$V_{Da,Ab}^{coul} = \frac{1}{4\pi\varepsilon_0} \int \frac{\rho_D^a(\vec{r_i})\rho_A^b(\vec{r_j})}{|\vec{r_i} - \vec{r_j}|} d\vec{r_i} d\vec{r_j} \tag{5}$$

where $\rho_D^a(\vec{r_i})$ is the transition density of excited state a of molecule D.

Computationally speaking, the transition density is expressed by an array of finite-size volume elements, which is called Transition Density Cube, from which the method takes the name. To

obtain $V_{Da,Ab}^{coul}$ we calculate the sum of the interactions of every couple of elements on the two Cubes, as in

$$V_{Da,Ab}^{coul} = \sum_{i,j} \frac{\rho_D^a(i) \cdot \rho_A^b(j)}{r_{ij}} \qquad (6)$$

where i denotes the set of 3D coordinates of the i-th point of the Cube of molecule D and r_{ij} is the distance between the two elements on the two cubes.

Within this approximation, the transition electric dipole moment associated to the Cube can be obtained as

$$\vec{\mu_a} = \sum_i \rho^a(i)\vec{r_i} \qquad (7)$$

thus the two transition electric dipole moments μ_a and μ_b can be retrieved from the two Cubes and the coupling and the orientation factor can be estimated also according to the PDA.

6 Additional Information

The data needed to run `JACoPO` are transition charges and transition density cubes. If you don't know how to obtain these data from quantum calculations with your favorite package, we suggest you read the manual of your program, have a look at the Multiwfn package (`https://multiwfn.codeplex.com/`) or contact the developers of your program for information.

7 Features and Options

A short help including the list of options is available by running the command

```
$ ./JACoPO.py -h
```

in the folder containing the JACoPO.py python file. The output of this command is as follows.

```
$ ./JACoPO.py -h
usage: JACoPO.py [-h] [--chg1 CHG1] [--chg2 CHG2] [--cub1 CUB1]
                 [--selcub1 SELCUB1 [SELCUB1 ...]] [--geo1 GEO1]
                 [--selgeo1 SELGEO1 [SELGEO1 ...]] [--cub2 CUB2]
                 [--selcub2 SELCUB2 [SELCUB2 ...]] [--geo2 GEO2]
                 [--selgeo2 SELGEO2 [SELGEO2 ...]] [--thresh THRESH]
                 [--coup {chgs,den}] [-o OUTPUT]

Calculates Electronic Coupling from Transition Charges and Densities.

optional arguments:
  -h, --help             show this help message and exit
  --chg1 CHG1            File with coordinates and charges for monomer 1.
                         (default: None)
  --chg2 CHG2            File with coordinates and charges for monomer 2.
                         (default: None)
  --cub1 CUB1            Transition Density Cube for monomer 1. (default:
                         mon1.cub)
  --selcub1 SELCUB1 [SELCUB1 ...]
                         Atom Selection for Transition Density Cube for monomer
                         1. This can either be a list or a file. (default:
                         None)
```

```
--geo1 GEO1             Geometry on which the Transition Density Cube of
                        monomer 1 will be projected. (Units: Angstrom)
                        (default: None)
--selgeo1 SELGEO1 [SELGEO1 ...]
                        Atom Selection for geometry 1. This can either be a
                        list or a file, which should contain the list.
                        (default: None)
--cub2 CUB2             Transition Density Cube for monomer 2. (default:
                        mon2.cub)
--selcub2 SELCUB2 [SELCUB2 ...]
                        Atom Selection for Transition Density Cube for monomer
                        2. This can either be a list or a file, which should
                        contain the list. (default: None)
--geo2 GEO2             Geometry on which the Transition Density Cube of
                        monomer 2 will be projected. (Units: Angstrom)
                        (default: None)
--selgeo2 SELGEO2 [SELGEO2 ...]
                        Atom Selection for geometry 2. This can either be a
                        list or a file. (default: None)
--thresh THRESH         Threshold for Transition Density Cubes. (default:
                        1e-05)
--coup {chgs,den}       Method of Calculation of the Electronic Coupling. If
                        no method is specified, both methods will be used.
                        (default: None)
-o OUTPUT, --output OUTPUT
                        Output File. (default: Coup.out)
```

Let's have a quick look at all these options and what they require as arguments.

- `--coup`: this option specifies the approach you want to use for the coupling calculation. You can choose between `chgs` and `den` for the Monopole–Monopole approximation or the TDC method respectively. The default value is not specified and calculation with both methods will be attempted.

- `--chg1,chg2`: these two options want filenames as argument. The file you should give is constituted by 5 columns. The first column contains the atomic symbols, the second to the fourth column contain the (x, y, z) coordinates of the atom (in Å), the last column contains the value of the charge of the atom (in a.u.). As default, these files are not specified. If those files are not found, calculation with the Monopole–Monopole approximation will be skipped.

- `--cub1,cub2`: these two options want filenames as argument. You should specify the transition density cube (Gaussian cube) files needed for the calculation with the TDC method. For general information regarding the structure of these files, have a look at `http://www.gaussian.com/g_tech/g_ur/u_cubegen.htm`. Please, make sure the coordinates of the structure specified in the appropriate section of the `.cub` files are expressed in atomic units. The default choice for these options are `mon1.cub` and `mon2.cub`, respectively.

- `--thresh`: this is a threshold to speed up TDC calculations. Elements of the cubes lower than this threshold will be ignored in the calculation of the coupling. For a more accurate, but slower, result, decrease this parameter.

- `-o,--output`: this is the name of the logfile `JACoPO` will write. The default is `Coup.out`.

Before explaining the other options, all related to the TDC method, a small digression is needed. A situation that could occur is that you want to evaluate the coupling for two equivalent

molecules in different positions in space. The value of the transition density is the same for these two molecules, what changes is just its position. What you need is a *reference* cube file that you want to project on the real geometry of the molecule in space.

The `--geo1,geo2` options allow doing this. More specifically, you should specify a file that is constituted by 4 columns: the first column contains the atomic symbols, the second to the fourth column contain the (x, y, z) coordinates of the atom (in Å). These files are basically Gaussian input files without the route section, and we assign them a `.inc` extension. If these files are specified, the cube selected with the `--cub1,cub2` option will be transformed in space to match the geometries specified with the `--geo1,geo2` option. If no other option is specified, `JACoPO` will assume that the atom order in the cube files is the same as in the geometry files.

If this is not the case, you can specify the atom correspondence with the `--selcub1,selcub2` and `--selgeo1,selgeo2` options. These options expect a list of atoms or a file, containing the list of atoms as it would be written if you were passing it directly. The rules to specify atom selections are simple and intuitive. Atoms should be separated by commas. For a more compact notation a group of consecutive atoms could be written in compact form (*e.g.* 3, 4, 5, 6 can be specified as 3–6).

Another realistic scenario is that your chromophore is included in a molecule, and the vicinities of the portion you are interested in are not UV active, *i. e.* they do not participate in the electronic excitations on your chromophore. In this case you want to project your cube on the chromophore, which is part of a bigger molecule. You can use the `--selcub1,selcub2` and `--selgeo1,selgeo2` options to specify the atoms you want to use for the transformation.

Examples of the files to be passed for each option are available in the `tests` folder.

8 Output

Examples of output files from `JACoPO` are in the `tests` folder. The first section of the output is a title and credits section.

```
##############
##  JACoPO  ##
##############


JACoPO: Just Another COupling Program, Obviously
JACoPO.py  Copyright (C) 2016  Daniele Padula, Marco Campetella
```

After this section, `JACoPO` writes information only on the calculations that have been chosen from the command line options. For the Monopole–Monopole approximation, an example is as follows.

```
###################################
##  Coupling Transition Charges  ##
###################################


Coupling calculated from transition charges according to
J. Phys. Chem. B, 2006, 110, 17268

Donor Structure and Charges:
  C      0.000000     1.394991     2.000000      0.095611
  C      1.208097     0.697495     2.000000      0.047806
  C      1.208097    -0.697495     2.000000     -0.047806
  C      0.000000    -1.394991     2.000000     -0.095611
  C     -1.208097    -0.697495     2.000000     -0.047806
  C     -1.208097     0.697495     2.000000      0.047806
  H      0.000000     2.494601     2.000000      0.063673
```

```
H      2.160388    1.247300    2.000000     0.031832
H      2.160388   -1.247300    2.000000    -0.031832
H      0.000000   -2.494601    2.000000    -0.063673
H     -2.160388   -1.247300    2.000000    -0.031832
H     -2.160388    1.247300    2.000000     0.031832


Donor Electric Transition Dipole Moment from Transition Charges in a.u.:
       x        y        z            norm
  0.0000   1.6566   0.0000           1.6566



Acceptor structure and Charges:
  C      0.000000    1.394991   -2.000000     0.095611
  C      1.208097    0.697495   -2.000000     0.047806
  C      1.208097   -0.697495   -2.000000    -0.047806
  C      0.000000   -1.394991   -2.000000    -0.095611
  C     -1.208097   -0.697495   -2.000000    -0.047806
  C     -1.208097    0.697495   -2.000000     0.047806
  H      0.000000    2.494601   -2.000000     0.063673
  H      2.160388    1.247300   -2.000000     0.031832
  H      2.160388   -1.247300   -2.000000    -0.031832
  H      0.000000   -2.494601   -2.000000    -0.063673
  H     -2.160388   -1.247300   -2.000000    -0.031832
  H     -2.160388    1.247300   -2.000000     0.031832


Acceptor Electric Transition Dipole Moment from Transition Charges in a.u.:
       x        y        z            norm
  0.0000   1.6566  -0.0000           1.6566


Orientation Factor:
1.00


Electronic Coupling according to PDA from Dipoles from Transition Charges in cm-1:
1394.54


Electronic Coupling in cm-1:
793.57
```

There is a small title section to remind you of the calculation that has been performed, followed by a reference containing the formula used. A recapitulation of the file specified with the `--chg1` option is reported, followed by the result of the calculation of the transition electric dipole moment for the geometry and charges you specified (see Eq. 4). An analogous section for the option `--chg2` is written.

Finally, the orientation factor (see Eq. 2) and two electronic couplings are reported. The first, is the result of the PDA applied on the dipoles calculated (see Eq. 1). The application points of the dipoles are the centers of mass of the geometries. The second is the coupling calculated with the Monopole–Monopole approximation (see Eq. 3). Please, be aware that the charges that you use for your input determine the quality of this calculation.

An example of a complete section for the TDC method is as follows.

```
####################################
##  Coupling Transition Densities  ##
####################################
```

```
Coupling calculated from transition densities according to
J. Phys. Chem. B, 1998, 102, 5378


Structure and Transition Density Cube in temp.cub moved to match geometry in 1.inc
RMSD (Ang):    0.0019

Donor Electric Transition Dipole Moment from Transition Density in a.u.:
      x        y        z            norm
   1.2234   1.7318   1.4028         2.5424


Structure and Transition Density Cube in temp.cub moved to match geometry in 2.inc
Atom correspondence between the cub and the geometry file, respectively:
   1   2   3   7   8   4   5  18   9  16   6  17  19  20  14  12  10  15  13  11  21
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
RMSD (Ang):    0.0019

Acceptor Electric Transition Dipole Moment from Transition Density in a.u.:
      x        y        z            norm
  -1.2234   1.7318  -1.4028         2.5424


Orientation Factor:
0.80

Electronic Coupling according to PDA from Dipoles from Transition Densities in cm-1:
595.87


Electronic Coupling in cm-1:
538.58
```

There is a small title section to remind you of the calculation that has been performed, followed by a reference containing the formula used. The first section refers to the cube specified with the option --cub1. When the --geo1 option is specified, a reminder of this option is printed and the RMSD (in Å, up to the 4th decimal digit) for the transformation is printed, to give you a rough idea of the quality of the transformation. If the RMSD is too high, maybe the selection of atoms is wrong, and you should check that. After this section, the result of the transition electric dipole calculation is reported (see Eq. 7).

Similar sections are present for the cube specified with the --cub2 option. In this case, since --selcub2 and --selgeo2 options have been specified, a summary of atom correspondence according to the options specified is also reported.

Finally, two electronic couplings are reported. The first, is the result of the PDA applied on the dipoles calculated (see Eq. 1). The application points of the dipoles are the centers of mass of the geometries. The second is the coupling calculated with the TDC method (see Eq. 6). Please, be aware that the cubes that you use for your input determine the quality of this calculation. In particular, for a better evaluation of the coupling, we recommend using thicker grids in the generation of the cube files.

At the end of the output file, a short summary of the calculations performed is reported.

```
#####################
##  Results Table  ##
#####################
```

```
 Calculation time: 00:01:16

 # Method          Coupling (cm-1)
 #------------------------------
 Tr Chgs            793.57
 PDA Dip Chgs      1394.54
 Tr Den            1405.30
 PDA Dip Den       1578.42
```

The calculation time and the results of calculations according to the required methods are summarized in a table.

Useful Commands

If you are not interested in reading the whole output file, the most straightforward command that you could use to have a quick look at the results is

```
$ tail Coup.out
####################

 Calculation time: 00:01:16

 # Method          Coupling (cm-1)
 #------------------------------
 Tr Chgs            793.57
 PDA Dip Chgs      1394.54
 Tr Den            1405.30
 PDA Dip Den       1578.42
```

If you are interested in extracting directly a number out of the output, you could use the command

```
$ grep "Tr Den" Coup.out | awk '{print $NF}'
1405.30
```

Substituting the string to search in the `grep` command gives you access to the result obtained with the method you prefer. This last command is useful for incorporation into shell scripts.

Acknowledgements

References

[1] T. Förster, *Ann. Physik*, 1948, **437**, 55–75.

[2] R. W. Woody, *J. Chem. Phys.*, 1968, **49**, 4797–4806.

[3] M. E. Madjet, A. Abdurahman and T. Renger, *J. Phys. Chem. B*, 2006, **110**, 17268–17281.

[4] B. P. Krueger, G. D. Scholes and G. R. Fleming, *J. Phys. Chem. B*, 1998, **102**, 5378–5386.