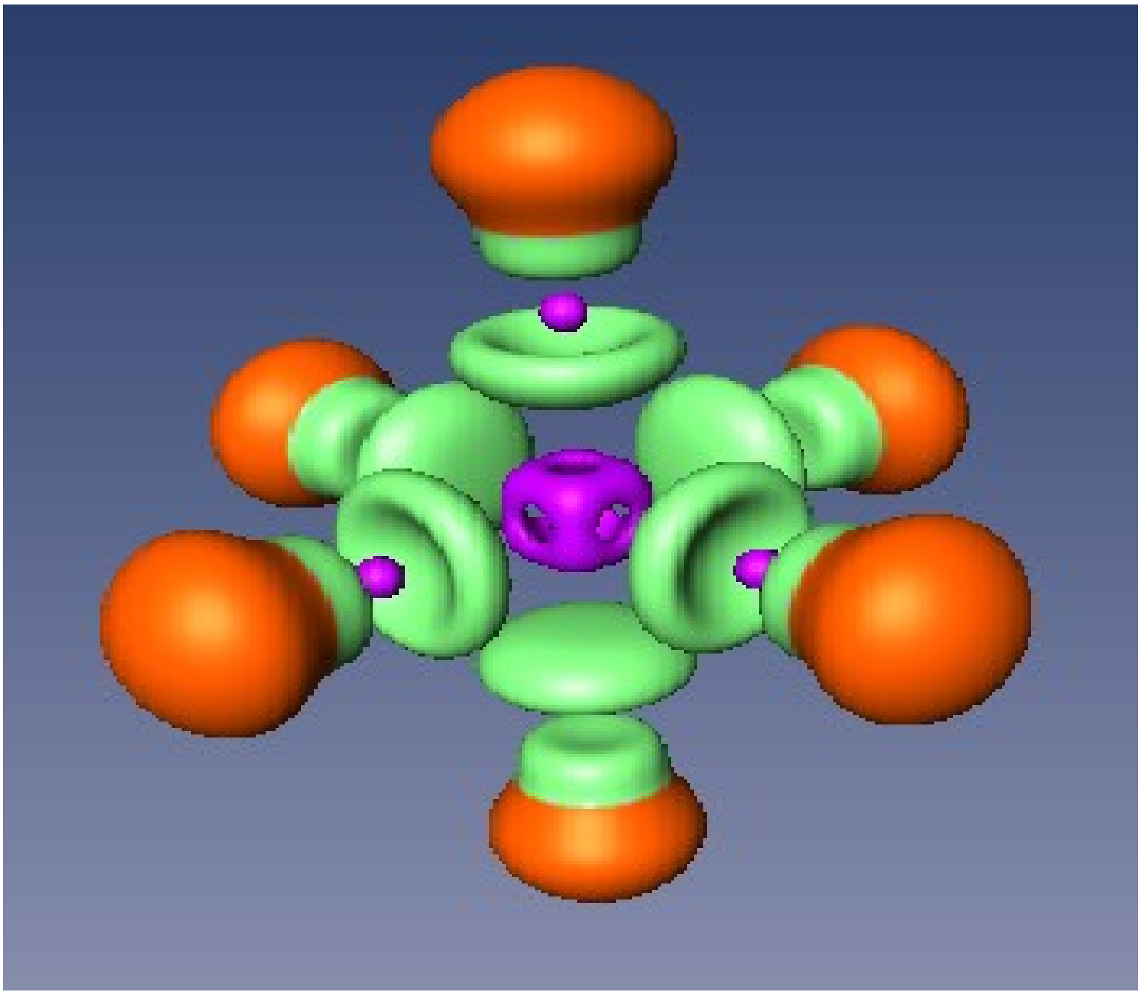
The TopMoD

user's manual



Bernard Silvi
Laboratoire de Chime Théorique
Université Pierre et Marie Curie

Contents

1 General Presentation

The ToPMoD package enables the calculation of the ELF function on a 3-dimensional grid, the assignment of the basins and the calculation of the basin populations and of their variance. It uses wave functions written in a `wfn` file which is available as output file generated by Gaussian92/94/98/03 and GAMESS *ab initio* softwares. The modules are written in FORTRAN 90 in order to enable dynamic memory allocation.

In order to carry out a standard ELF analysis three programs have to be run in the following order:

- grid90: calculates ELF on a 3-D grid parallel to the standard axis defined in the MO calculation (this allows to exploit the factorization of the gaussian functions),
- bas90: assigns the grid points to basins,
- pop90: calculates the basin populations and variances.

The other modules do the following tasks:

- top_sym: exploits abelian symmetry operations,
- search90: localizes the critical points of the ELF gradient field,
- mod_wfn: enables to remove atoms in the case of large systems in order to focuss the calculation on the region of interest,
- sym_wfn: symmetrizes the wfn file for $^2\Pi$ states of linear molecules
- bas_to_syn: assigns basin types from `bas.sbf` file,
- sbf_to_cube: converts `sbf` files to `cube` files for Molekel,
- sbf_to_am: converts `sbf` files to i`am` files for Amira,
- wfn_to_line: generates a wireframe, stick or ball and stick molecular skeletons for Amira.

1.1 Useful reading and references

References
A. Becke and K. E. Edgecombe,*J. Chem. Phys.*, **92**, 5397–5404 (1990)
B. Silvi and A. Savin, *Nature*, **371**, 683–686 (1994)
A. Savin, B. Silvi and F. Colonna, *Can. J. Chem.*, **74**, 1088-1096 (1996)
X. Krokidis, S. Noury and B. Silvi, *J. Phys. Chem. A*, **101**, 7277–7282 (1997)
B. Silvi, A. Savin, and F.R. Wagner, In *Modeling of Minerals and Silicated Materials*, B. Silvi and Ph. D'Arco Eds., Kluwer Academic Publisher, Dordrecht, 1997
S. Noury, F. Colonna, A. Savin and B. Silvi, *J. Mol. Struct.*, **450**, 59–68 (1998)
please cite:
S. Noury, X. Krokidis, F. Fuster and B. Silvi, TopMod package, 1997.
S. Noury, X. Krokidis, F. Fuster and B. Silvi, *Computers and Chemistry,* **23**, 597–604 (1999)
M. Calatayud, J. Andrés, A. Beltrán and B. Silvi, *Theoret. Chem. Acc.*, **2001**; **105**, 299 (2001)
B. Silvi, *J. Mol. Struct.*, **614**, 3 (2002)
B. Silvi, *J. Phys. Chem. A,* **107**, 3081 (2003)
B. Silvi, *Phys. Chem. Chem. Phys.,* **6**, 256 (2204)
E. Matito, B. Silvi, M. Duran and M. Solà, *J. Chem. Phys.,* **125**, 024301 (2006)

please report bugs or problems via electronic mail at:
silvi@lct.jussieu.fr

2 Gaussian and GAMESS inputs

The `wfn` files are written by GAMESS and by Gaussian. It is not a standard option and therefore it must be indicated in the input.

2.1 GAMESS

Set `AIMPAC=.TRUE.` in the `$ CONTRL` line command.

Example

```
$ CONTRL SCFTYP=RHF RUNTYP=OPTIMIZE MPLEVL=0 ICHARG=0 MULT=1 COORD=ZMT
AIMPAC=.TRUE. $ END
$ SYSTEM TIMLIM=100 MEMORY=800000 $ END
$ BASIS GBASIS=n31 NGAUSS=6 NDFUNC=1 NPFUNC=1 $ END
$ GUESS GUESS=HUCKEL $ END
$ DATA
CH3OH
Cs

H
O 1 oh
C 2 co 1 coh
H 3 ch1 2 hco1 1 180.0
H 3 ch2 2 hco2 1 t
H 3 ch2 2 hco2 1 -t

oh=0.979
co=1.3
coh=110.0
ch1=1.0
hco1=109.0
ch2=1.0
hco2=109.0
t=60.0

$ END
```

2.2 Gaussian

Set `out=wfn` in the Route section and give the name of the `wfn` file at the end of the molecule specification. The title first word (contiguous non blaink characters) are used to form the file names in, it might be dangerous to use non alpha numerical characters.

2.2.1 Hartree Fock or DFT calculations

Example: C_4H_4 tetrahedrane

```
# P HF/6-311++G(2df,2p) opt out=wfn

tetrahedrane

0,1
C -0.524831 0.524831 0.524831
C 0.524831 -0.524831 0.524831
C -0.524831 -0.524831 -0.524831
C 0.524831 0.524831 -0.524831
H -1.134794 1.134794 1.134794
H 1.134794 -1.134794 1.134794
H -1.134794 -1.134794 -1.134794
H 1.134794 1.134794 -1.134794

tetrahedrane.wfn
```

To compute condensed Fukui functions over basins it is necessary to set
tt IOP(99/18=1) such in the following example:

$BeCl_2$: input for condensed Fukui function calcualtion

```
#P B3LYP/6-31G** popt iop(99/18=1) out=wfn

BeCl2

0, 1
Cl
Be 1 r
X 2 1.0 1 90.0
Cl 2 r 3 90.0 1 180.0

r=1.62

becl2.wfn
```

2.2.2 Post Hartree Fock calculations

The `NaturalOrbitals` or `NaturalSpinOrbitals` should be specified as options for the population keyword and `all` in the density one.

Closed shell example: water molecule

```
#CISD/6-31G** popt pop=no density=all out=wfn

H2O

0 1
H
O 1 r
H 2 r 1 a

r=1.0
a=110.0

h2ono.wfn
```

Open shell example: CH_3 radical

```
#P CISD/6-31G** popt pop=NaturalSpin density=all out=wfn

CH3no

0 2
H
C 1 r
H 2 r 1 120.0
H 2 r 1 120.0 3 180.0

r=1.07

ch3no.wfn
```

3 grid09 input

1. `filein` (a40) : wfn file name

2. `origin(3)` (free format) : coordinates of the origin of the box defining the grid

3. `edge(3)` (free format) : length of the edges of the box in Bohr

4. `intervalx, intervaly, intervalz` (free format) : number of intervals on each edge. Choose the number of intervals in order to have steps of about 0.1 a.u and to avoid grid points in a symmetry plane.

in output the program generates title_elf.sbf, title_rho.sbf and itle_lap.sbf files)in which title is the first non-blank characters of the title in gaussian input (here water)

3.1 Input example: water molecule

The `h2o.wfn` file begins by the following lines:

```
H2O
GAUSSIAN 5 MOL ORBITALS 30 PRIMITIVES 3 NUCLEI
O 1 (CENTRE 1) 0.00000000 0.00000000 0.22212920 CHARGE = 8.0
H 2 (CENTRE 2) 0.00000000 1.43441917 -0.88851679 CHARGE = 1.0
H 3 (CENTRE 3) 0.00000000 -1.43441917 -0.88851679 CHARGE = 1.0
```

The $x = 0$ and $y = 0$ planes can be used as symmetry planes

3.1.1 grid09 input without symmetry

```
h2o.wfn
-5.0 -6.0 -6.0
10.0 12.0 12.0
100 120 120
```

The program will write a 101 x 121 x 121 grid on the `H2O_elf.sbf` file to be directly used by bas09for the ELF analysis `H2O_rho.sbf` file to be directly used by bas09 for the AIM analysis `H2O_lap.sbf` file to display the laplacian of the charge density.

3.1.2 grid09 input with 2 symmetry planes

```
h2o.wfn
0.05 0.05 -6.0
5.0 6.0 12.0
50 60 120
```

The program will write a 51 x 61 x 121 grid on the `H2O_elf.sbf` file to be updated by top_sym.

4 bas09 input

1. `function` (a3) : type of function elf/rho

2. `filein` (a40) : input wfn file

3. `iacc` free format : accuracy level iacc=0 no approximation in derivative evaluation, 1 and 2 use of a cutoff. 1 is recommanded.

4. `external core shell option` (a1) : do not merge the external core shell basins (y/n)

5. for each atom bearing a pseudopotential

   - `ns1` (free format) : type of the pseudopotential 1 large core, 2 small core

6. `attractor search mode` (free format) :

   - < 0 fast automatic from the current grid, finds only attractors within the box.
   - 0 full automatic, search in the whole molecular space (recommended if symmetry is used).
   - > 0 number of expected non protonated valence basins given in input
   - skip if automatic search, for each attractor in input:
     - `xp, yp, zp` (free format) : cartesian coordinates of the attractor (in a. u.)
     - `type, ref_atom`

       `type`: 2 point, 3 circle, 4 sphere.

       `ref_atom` is the number of the atom used to define the C_∞ axis in case of circular attractor. For point and sphere attractor put any integer.
     - `s_order`: synaptic order of the basin.
     - `radius`: radius of the circle or of the sphere, for point attractors 0.5 can be used safely.
     - basin_name (3a4): atomic symbols used to name the basin.

7. `assign grid points` (a3) : (y/n)

In output the program writes one file: title_ebas.sbf or title_ras.sbf according to the chosen function to be used by pop09 and one input file for bas09 named temp.bas. The temp.bas file corresponds to the input attractor mode, it can be generated with a different wave function (for example with a smaller basis set) which nevertheless yields the same overall topology and without grid point assignment. In this case it is not necessary to run grid09 before. This procedure is recommended for large systems. Do not forget to update the wfn file name in temp.bas if another wfn file is used.

4.1 Example: H_2O all electron calculation

```
Fast automatic search              from input
elf                                elf
h2o.wfn                            h2o.wfn
1                                  1
n                                  n
0                                  2
                                   1.0 0.0 0.7
                                   2 0
                                   1
                                   0.5
                                   O
                                   -1.0 0.0 0.0
                                   2 0
                                   1
                                   0.5
                                   O
y                                  y
```

4.2 Example: H_2O pseudopotential (large core) calculation

```
Automatic search                   from input
elf                                elf
h2o.wfn                            h2o.wfn
1                                  1
n                                  n
0                                  1
-1                                 1
                                   2
                                   1.0 0.0 0.7
                                   2 0
                                   1
                                   0.5
                                   O
                                   -1.0 0.0 0.0
                                   2 0
                                   1
                                   0.5
                                   O
y                                  y
```

5 top_sym input

1. `file_in` (a40) : sbf input file
2. `nop` (free format) : number of symmetry operations (≤ 3)
3. `nop` times
   (a) `name_op` (a3) : symmetry operation symbol (see table)
   (b) `xc, yc, zc` (free format) : coordinates defining the operation (see table)

symbol	operation	`xc, yc, zc`
`INV`	inversion	center of inversion coordinates
`PLX`	$x = xc$ plane	$xc, 0.0, 0.0$
`PLY`	$y = yc$ plane	$0.0, yc, 0.0$
`PLZ`	$z = zc$ plane	$0.0, 0.0, zc$
`PXY`	$y = x + yc$ plane	$0.0, yc, 0.0$
`PYX`	$y = -x + yc$ plane	$0.0, yc, 0.0$
`PXZ`	$z = x + zc$ plane	$0.0, 0.0, zc$
`PZX`	$z = -x + zc$ plane	$0.0, 0.0, zc$
`PYZ`	$z = -y + zc$ plane	$0.0, 0.0, zc$
`PZY`	$z = y + zc$ plane	$0.0, 0.0, zc$
`C2X`	$C_2(x)$ axis	$0.0, yc, zc$ intersection with $x = 0.0$ plane
`C2Y`	$C_2(y)$ axis	$xc, 0.0, zc$ intersection with $y = 0.0$ plane
`C2Z`	$C_2(z)$ axis	$xc, yc, 0.0$ intersection with $z = 0.0$ plane
`C4X`	$C_4(x)$ axis	$0.0, yc, zc$ intersection with $x = 0.0$ plane
`C4Y`	$C_4(y)$ axis	$xc, 0.0, zc$ intersection with $y = 0.0$ plane
`C4Z`	$C_4(z)$ axis	$xc, yc, 0.0$ intersection with $z = 0.0$ plane
`S4X`	$S_4(x)$ axis	xc, yc, zc center of inversion coordinates
`S4Y`	$S_4(y)$ axis	xc, yc, zc center of inversion coordinates
`S4Z`	$S_4(z)$ axis	xc, yc, zc center of inversion coordinates
`CTX`	$C_4(x)$ screw axis	yc, zc intersection, xc translation with $x = xc$ plane
`CTY`	$C_4(y)$ screw axis	xc, zc intersection, yc translation with $y = yc$ plane
`CTZ`	$C_4(z)$ screw axis	xc, yc intersection, zc translation with $z = zc$ plane

Symmetry operations symbols and coordinates.

5.1 Example

```
H2O_bas.sbf
2
PLX
```

```
0.0 0.0 0.0
PLY
0.0 0.0 0.0
```

The program will write a 101 x 121 x 121 grid on the `H2O_bas.sbf` which is now updated.

Case of T_d group In the case of the T_d group, the best way to exploit symmetry is to put the vertices of the tetrahedron at the vertices of a cube like in the following example:

```
C 1 (CENTRE 1) 0.98523954 0.98523954 0.98523954 CHARGE = 6.0
C 2 (CENTRE 2) -0.98523954 -0.98523954 0.98523954 CHARGE = 6.0
C 3 (CENTRE 3) -0.98523954 0.98523954 -0.98523954 CHARGE = 6.0
C 4 (CENTRE 4) 0.98523954 -0.98523954 -0.98523954 CHARGE = 6.0
```

The input for grid09 corresponds to a qurter of a cube, i.e.:

```
0.05 0.05 -7.25
7.20 7.20 14.5
```

Then run top_sym with two $C2$ rotation, one around the x axis, the other around y:

```
tetrahedrane_ebas.sbf 2
C2X
0.0 0.0 0.0
C2Y
0.0 0.0 0.0
```

6 pop09 and fast_pop input

1. `filein` (a40) : input wfn file
2. `tol` (free format): integration threshold (recommended value 7)
3. `n_elf, n_aim` (free format) : numbers of ELF and AIM basins considered (if 0 all the basins are taken into account in the calculation)
4. skip if `n_elf=0`: `(elf_ind(i),i=1,n_elf)`: label of the considered ELF basins in the order of the bas09 output.
5. skip if `n_aim=0`: `(aim_ind(i),i=1,n_aim)`: label of the considered AIM basins in the order of the bas09 output.

6.1 Example: H_2O all electron calculation

Full calculation

```
h2o.wfn
7
0 0
```

only V(H2,O) and Atom(O)

```
h2o.wfn
7
1 1
3
2
```

The variance calculation is not valid

for MP2 wave functions use fast_pop. The variance calculation is not valid on a part of the molecule use fast_pop (asymmetric unit).

6.2 pop09 output: definition of calculated properties.

The following symbols have been used to define the calculated properties:

1. Ω_i: basin labelled by i
2. $\rho(\mathbf{r})$: spinless electron density.
3. $\rho^{\alpha}(\mathbf{r})$: α-spin electron density.
4. $\rho^{\beta}(\mathbf{r})$: β-spin electron density.
5. $\pi(\mathbf{r},\mathbf{r}')$: spinless electron pair density.
6. $\pi^{\alpha\beta}(\mathbf{r},\mathbf{r}')$: $\alpha\beta$ contribution to electron pair density.
7. $\pi^{\alpha\alpha}(\mathbf{r},\mathbf{r}')$: $\alpha\alpha$ contribution to electron pair density.
8. $\pi^{\beta\beta}(\mathbf{r},\mathbf{r}')$: $\beta\beta$ contribution to electron pair density.

6.2.1 Basin properties.

1. basin volume $V(\Omega_i) = \int\limits_{\Omega_i} d\mathbf{r}$, for $ELF \geq 0.02$
2. basin population $\bar{N}(\Omega_i) = \int\limits_{\Omega_i} \rho(\mathbf{r})d\mathbf{r}$.
3. $\alpha\beta$ pair population $\bar{P}^{\alpha\beta}(\Omega_i, \Omega_i) = \int\limits_{\Omega_i}\int\limits_{\Omega_i} \pi^{\alpha\beta}(\mathbf{r}, \mathbf{r}')d\mathbf{r}d\mathbf{r}'$.
4. $\alpha\alpha$ pair population $\bar{P}^{\alpha\alpha}(\Omega_i, \Omega_i) = \int\limits_{\Omega_i}\int\limits_{\Omega_i} \pi^{\alpha\alpha}(\mathbf{r}, \mathbf{r}')d\mathbf{r}d\mathbf{r}'$.
5. $\beta\beta$ pair population $\bar{P}^{\beta\beta}(\Omega_i, \Omega_i) = \int\limits_{\Omega_i}\int\limits_{\Omega_i} \pi^{\beta\beta}(\mathbf{r}, \mathbf{r}')d\mathbf{r}d\mathbf{r}'$.
6. variance $\sigma^2(\Omega_i) = \int\limits_{\Omega_i}\int\limits_{\Omega_i} \pi(\mathbf{r}, \mathbf{r}')d\mathbf{r}d\mathbf{r}' + \bar{N}\Omega_i - \bar{N}^2(\Omega_i)$.
7. relative fluctuation $\lambda(\Omega_i) = \sigma^2(\Omega_i)/\bar{N}\Omega_i)$
8. integrated spin density $S_z(\Omega_i) = \frac{1}{2}\int\limits_{\Omega_i} \left(\rho^\alpha(\mathbf{r}) - \rho^\beta(\mathbf{r})\right) d\mathbf{r}$.
9. condensed Fukui functions f^+, f^0, f^-

$$\begin{aligned}
\langle f^- \rangle &= \int\limits_{\Omega_i} \left[\rho_N(\mathbf{r}) - \rho_{N-1}(\mathbf{r})\right] d\mathbf{r} \\
\langle f^+ \rangle &= \int\limits_{\Omega_i} \left[\rho_{N+1}(\mathbf{r}) - \rho_N(\mathbf{r})\right] d\mathbf{r} \\
\langle f^0 \rangle &= \frac{1}{2}\left(\langle f^- \rangle + \langle f^+ \rangle\right)
\end{aligned}$$

   To get condensed Fukui set `IOP(99/18=1)` in Gaussian03 input file

6.2.2 variance related properties.

1. α-spin entropy: $\sum\limits_{i,j} p^{\alpha\alpha}(\Omega_i, \Omega_j) \log \frac{p^{\alpha\alpha}(\Omega_i,\Omega_j)}{n^\alpha(\Omega_i)n^\alpha(\Omega_j)}$.
2. β-spin entropy: $\sum\limits_{i,j} p^{\beta\beta}(\Omega_i, \Omega_j) \log \frac{p^{\beta\beta}(\Omega_i,\Omega_j)}{n^\beta(\Omega_i)n^\beta(\Omega_j)}$.
3. total entropy: $\sum\limits_{i,j} p(\Omega_i, \Omega_j) \log \frac{p(\Omega_i,\Omega_j)}{n(\Omega_i)n(\Omega_j)}$.
4. covariance $B_{i,j} = \bar{N}(\Omega_i)\bar{N}(\Omega_j) - P(\Omega_i, \Omega_j)$.
5. relative covariance $B_{i,j}/\bar{N}(\Omega_j)$.

6. fluctuation contribution: $B_{i,j}/\sum\limits_{j\neq i} B_{i,j}$

where $p(\Omega_i,\Omega_j) = P(\Omega_i,\Omega_j)/N(N-1), n(\Omega_i) = \bar{N}(\Omega_j)/N$.

7 search09 input

1. `filein` (a40) : input wfn file
2. `type of critical point` (free format) : 0 any, 1 attractor, -1 end
3. `x, y, z` (free format) : guessed location

7.1 Example

```
h2o.wfn
0
1.0 -1.  0.5
1
0.3 0.2 0.1
-1
```

8 wfn file manipulations

For some specific appliacations the `wfn` file must be modified because GAUSSIAN contains a bug in the case of ROHF calculations (all the orbitals are doubly occupied!). There are two `wfn` manipulations:

1. ROHF: change occupation number of singly occupied orbitals

   ```
   MO   5   MO 0.0   OCC NO = 2.0000000  ORB. ENERGY =   0.0

   MO   5   MO 0.0   OCC NO = 1.0000000  ORB. ENERGY =   0.0
   ```

2. ROHF and CASSCF: specify orbital contribution to M_S (in fact twice). The 0.0 after MO should be changed to -1 (β), 0 (mixing true singlet), 1 (α), example

   ```
   MO   5   MO 1.0   OCC NO = 1.0000000  ORB. ENERGY =   0.0
   ```

8.1 sym_wfn

Molecular *ab initio* programs use basis real functions, in the case of linear molecules these computed MOs are not eigenfunctions of the L_z operator and therefore the cylindrical symmetry can be lost. This is particularly the case for open shell systems in a Π state. `sym_wfn` as been wriiten to correct this drawback for $^2\Pi$ states calculated with ROHF. In order to get symmetrized figures run `sym_wfn` before performing the grid09 calculation. The program prompts the user to get the input `wfn` file name and the name of the symmetrized `sym_wfn` file.

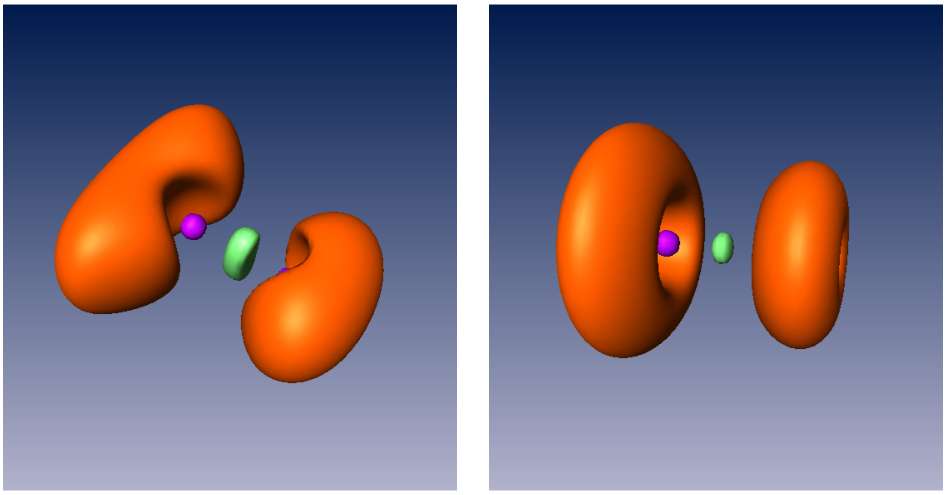


Figure 1: NO ($^2\Pi$): left unsymmetrized, right symmetrized

8.2 mod_wfn

For large molecular systems `mod_wfn` enables to restrict the analysis to a subpart of the system. The atoms of the subpart of interest and their first neighbours are entered as data of `mod_wfn`

8.2.1 mod_wfn input

1. `filein` (a40) : wfn file name
2. `natoms2` (free format) : number of selected atoms
3. `iatom(i),i=1,natoms2` : list of the selected atoms
4. `fileout` (a40) : modified wfn file name

Example: CH_2ClCH_2COOH The selected subpart is the functional group. The atoms in the wfn appear in the following order:

```
C     1     (CENTRE  1)   1.52388347  0.82729056  0.60836370  CHARGE =  6.0
C     2     (CENTRE  2)  -0.36562060 -1.02000099 -0.54662559  CHARGE =  6.0
Cl    3     (CENTRE  3)   4.74167762 -0.16320636 -0.05397264  CHARGE = 17.0
C     4     (CENTRE  4)  -3.10102364 -0.30181750 -0.07030471  CHARGE =  6.0
O     5     (CENTRE  5)  -4.78491297 -1.77008583  0.40373635  CHARGE =  8.0
O     6     (CENTRE  6)  -3.48476780  2.22384618 -0.29263556  CHARGE =  8.0
H     7     (CENTRE  7)   1.35335617  0.91475856  2.66069875  CHARGE =  1.0
H     8     (CENTRE  8)   1.30958103  2.72008831 -0.16895560  CHARGE =  1.0
```

```
  H     9     (CENTRE  9)   -0.08789824 -1.09332176 -2.59892348   CHARGE =   1.0
  H    10     (CENTRE 10)   -0.09155624 -2.92840929  0.18473731   CHARGE =   1.0
  H    11     (CENTRE 11)   -5.27799149  2.49847708  0.00257117   CHARGE =   1.0
```

The selected atoms are C(2), C(4), O(5), O(6) and H(11). The input is :

```
ch2clch2cooh.wfn
 5
 2 4 5 6 11
ghost.wfn
```

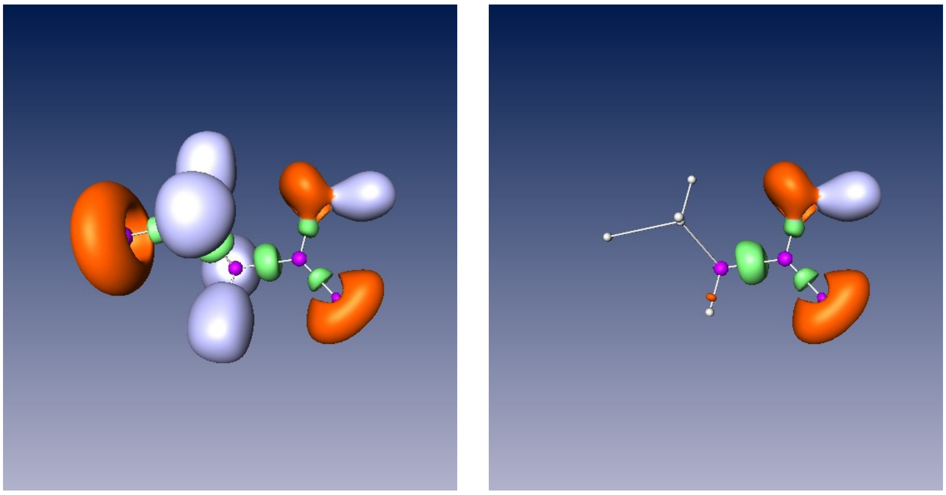


Figure 2: Display of CH_2ClCH_2COOH localization domains. Left: total, right: selection.

9 Visualization

In order to visualize, first create a color code file with the bas_to_syn utility program. According to your data analyzer, we recommend Amira and Molekel (version 4), generate either title_elf.cube and title_esynf.cube files with the sbf_to_cube utility for Molekel. The sbf_to_am utility converts the sbf files to the am format readable by Amira.

Molekel is an Avanced Interactive 3D-graphics package for visulizing molecular and electronic structure data from output of various Quantum Chemistry softwares such as GAUSSIAN94/98, GAMESS-US,ADF, HONDO. It works under UNIX, LINUX and also WINDOWS. It uses the Open/GL, Mesa, GLUT and GLUI libraries. MOLEKEL was designed and developed at University of Geneva and CSCS / ETHZ by Peter F. Flükiger in the early 90ties and later by Stefan Portmann.

Molekel is available at the following URL:
http://www.cscs.ch/molekel

A `ppt` file explains how to use Molekel. The `syn.rgb`file in the `SYN` directory provides the texture and color codes.

Amira was originally distributed by TGS. The following URL provides useful piece of information and enable to download a documentation brochure and a free trial
`http://www.amiravis.com/`

The `syn.am` file in the `SYN` directory provides the texture and color codes.

9.1 sbf_to_cube and sbf_to_am inpiuts

9.1.1 sbf_to_cube

1. `filein` (a40) : wfn file name
2. `filesbf`(a40) : sbf file to be converted
3. `ix,iy,iz` : step size in each direction (standard 1 1 1)

Run sbf_to_cube for the elf and syn sbf files.

9.1.2 sbf_to_am

1. `filesbf`(a40) : sbf file to be converted

Run sbf_to_am for the elf and syn sbf files.