Carbene Stabilization by Aryl Substituents

The most updated version of this handout can always be found at:

`https://github.com/duartegroup/sbmcc/blob/master/carbene/sbmcarbene.pdf`

Many of the calculations described in the paper[1] have already been covered in the Day 1 tutorial, so here we will only cover "new" procedures which may be of interest. Much of this information can be found in the ORCA manual and is documented in much greater detail there.

1 Dispersion interactions

Dispersion interactions[2] can (and should) be accounted for by adding the keyword `D3` to your input file (after the functional and basis set). This keyword is only available for certain functionals: see `https://www.chemie.uni-bonn.de/pctc/mulliken-center/software/dft-d3/functionalsbj` for a full list of the functionals for which it can be used with. As long as a functional is listed there, it will work with the `D3` keyword in ORCA: it doesn't matter whether the "statistical data" is available or not. The time taken for the calculation with or without dispersion is virtually the same, but more physically accurate results are obtained.

2 Visualising orbitals

Virtually every calculation that ORCA carries out involves the calculation of some sort of molecular orbitals. In order to visualise them, you will first need to download a free trial of ChemCraft (which lasts for 150 days): `https://chemcraftprog.com`. (Avogadro is extremely buggy in this respect and often crashes when trying to visualise orbitals.) In addition, in the input file you will need to add the two extra keywords:

```
PrintBasis PrintMOs
```

to the first line of your input file. If you have already performed a long optimisation and do not want to re-run it with the new keywords, you can simply run a single-point calculation using the optimised geometry (remember to also use the same level of theory).

Open the `.out` file using ChemCraft (you cannot drag the file into the programme, so you have to use **File** > **Open**). At this point you can change the colour scheme (if desired...) with **Display** > **CPK coloring 1 (dark background)**, or any of the other alternatives. To visualise the orbitals, go to **Tools** > **Orbitals** > **Render molecular orbitals**.

ChemCraft will give you a list of the MOs calculated, labelled by their occupancy (i.e. how many electrons are in them) and energies (in hartrees). Select all the orbitals of interest, increase *Map points per angstrom* to around 12, and click **Ok**. Typically, we are only interested in the orbitals near the HOMO and LUMO: very low-energy occupied orbitals are usually just core orbitals such as carbon 1s.

Once ChemCraft finishes its calculations, select any of the MOs from the left-hand pane that appears. Turn the *contour value* up to around 0.10, then click on both **Show isosurface** and **Both-signed**.

3 NBO calculations

The canonical MOs, which are routinely calculated by ORCA, are typically delocalised over many atoms in the molecule and do not bear much resemblance to σ or π MOs of individual functional groups. These can be decomposed (more accurately, transformed) into *natural bond orbitals* (NBOs),[3] which attempt to localise the MOs in a way which allows one-to-one correspondence with localised σ or π bonds.

In order to perform this localisation, a separate NBO programme is required. Unfortunately this is not installed on the `arcus-b` cluster (and is not free to download), so if you want to run these calculations, please consult a demonstrator. Nevertheless, the steps will be briefly documented here.

Firstly, the environment variable `NBOEXE` has to be set to the full path of the NBO programme. This is best done inside `qorca` by adding the line

```
print('export NBOEXE=/path/to/nbo.exe', file=bash_script)
```

at the appropriate location. This ensures that ORCA knows where the NBO programme resides in, so that ORCA can call it to perform its tasks. Requesting an NBO analysis then simply consists of adding the `NBO` keyword to the input file. Near the bottom of the input file, the NBO programme will print its output. This includes a description of the localised MOs found (whether they are core MOs, MOs localised to a bond, multiple-centre MOs, or lone pairs), as well as the “amount” of atomic orbitals that go into each MO. Therefore, if a lone pair has 33% carbon 2s character and 67% carbon 2p character, this means that it is a sp^2-hybridised lone pair. Likewise, one can identify the hybrid orbitals that form individual bonds.

4 NICS calculations

The *nucleus-independent chemical shift* method is one way of assessing the aromaticity of a ring.[4] In order to calculate this parameter, we need to add a dummy atom in the middle of the ring of interest. Dummy atoms can be added to the coordinate block in an ORCA input file using the label `DA`. The overall input file will look like this:

sample_nics.inp

```
! FUNC BASIS AUXBASIS

%pal nprocs 16 end

*xyz 0 1
 ... (original coordinates go here) ...
 DA    X_DA    Y_DA    Z_DA
*

%eprnmr
  Ori GIAO
  Nuclei = DA_NUMBER { shift };
end
```

There are a few new things in this input file. We will go through them in sequence. You should be comfortable with `FUNC` and `BASIS`, but `AUXBASIS` is probably new to you. This refers to an auxiliary basis set, which is used by ORCA when approximating certain integrals in the NMR calculation. For now, it suffices to follow this table when choosing `AUXBASIS`:

`BASIS`	Appropriate `AUXBASIS`
`def2` family	`def2/JK`
`cc-pVnZ` family	`cc-pVnZ/JK`
Anything else	`AutoAux`

The line beginning with `DA` simply specifies the dummy atom as well as its Cartesian coordinates, just like how every other atom is specified. The coordinates of the dummy atom are simply the mean of the coordinates of every heavy atom in the ring:

$$x_{\text{DA}} = \frac{1}{6}(x_{\text{C1}} + x_{\text{C2}} + \cdots + x_{\text{C6}})$$

To automate this calculation, first download some helper scripts using:

```
source <(wget -O - https://raw.githubusercontent.com/duartegroup/sbmcc/master/carbene/nicsdl.sh)
```

This will download two Python scripts to your home directory. `cd` to there and run

```
python nics1.py
```

At the prompt, paste in the coordinates of the six carbons that form the benzene ring (that should be six lines' worth of `C NUMBER NUMBER NUMBER`). The script will output a new line starting with `DA`, which you should then copy and paste back into the geometry in your input file.

Last comes the `%eprnmr` block. `GIAO` stands for *gauge-independent atomic orbitals* (or *including* or *invariant*; chemists can't seem to make up their minds). This is the most widely-used method for calculating NMR chemical shifts (more information on this can be found in the recommended books). The line `Nuclei = DA_NUMBER { shift };` specifies that we only want the chemical shift of the dummy atom (you *can* calculate other chemical shifts if you want, but it will take a long time). Replace `DA_NUMBER` with the appropriate atom number corresponding to the dummy atom.

Warning

Within the `%eprnmr` block, atom labels are counted **beginning from 1, not 0**. So if your dummy atom is the fifth atom to be listed, you should replace `DA_NUMBER` with 5, not 4.

Confusingly, in the *output*, the atom numbering returns to "normal" and is counted from 0 again. When the calculation finishes, there will be a section at the bottom of the output file containing a `CHEMICAL SHIELDING SUMMARY`. Just above that is the `Total shielding tensor`, which is the quantity of interest here. The NICS value corresponds to one of the three *principal values* of this tensor, with signs reversed. Helpfully, ORCA will calculate these three values, and they are the first three numbers in the `Total` line under `Diagonalised sT*s matrix`. But unhelpfully, ORCA doesn't tell you *which* of the three principal values is the one we are actually interested in.

So, we have to do that calculation ourselves. This involves finding the *principal axes* of the shielding tensor, and identifying which of these three axes is perpendicular to the benzene ring. There is also a helper script for that, which can be called from your home directory with

```
python nics2.py
```

At the first prompt, paste in the three lines under `Total shielding tensor`; and at the second prompt, paste in the coordinates of *any* three atoms from the ring of interest. The script will then calculate which principal value is the correct one to use: make sure that the number it gives you at the end is the same as one of the principal values ORCA calculated (except with reversed sign). This number is the NICS(0)$_{zz}$ number: a large negative value indicates aromaticity, whereas a large positive value indicates antiaromaticity.

For the mathematically inclined, the principal axes are the eigenvectors of $\mathbf{s}^{\intercal}\mathbf{s}$ ($\mathbf{s}$ being the shielding tensor). Given a matrix $\mathbf{U}$ with these eigenvectors as its columns, the principal values are the diagonal elements of $\mathbf{U}^{\intercal}\mathbf{s}\mathbf{U}$. We have to find the principal axis which is perpendicular to the plane, and the vector normal to a plane can be found using $(\mathbf{c}_1 - \mathbf{c}_2) \times (\mathbf{c}_1 - \mathbf{c}_3)$, where the $\mathbf{c}_i$'s are vectors from the origin to different points in the plane (which must not be collinear). Thus, to identify the correct principal axis, we simply need to see which principal axis most closely matches the normal vector (up to a constant factor), which is most easily done by comparing the dot products of each principal axis with the normal vector.

References

1. Woodcock, H. L.; Moran, D.; Brooks, B. R.; Schleyer, P. v. R.; Schaefer, H. F. *J. Am. Chem. Soc.* **2007**, *129*, 3763–3770.
2. Grimme, S.; Ehrlich, S.; Goerigk, L. *J. Comput. Chem.* **2011**, *32*, 1456–1465.
3. Weinhold, F.; Landis, C.; Glendening, E. *Int. Rev. Phys. Chem.* **2016**, *35*, 399–440.
4. Chen, Z.; Wannere, C. S.; Corminboeuf, C.; Puchta, R.; Schleyer, P. v. R. *Chem. Rev.* **2005**, *105*, 3842–3888.