

## 9.2.6 orca\_blockf

The `orca_blockf` utility program canonicalizes orbitals within a `.gbw` file for arbitrary subspaces. In this context, canonicalization refers to the block-diagonalization of the Fock matrix within the specified orbital subspace.

To run `orca_blockf` on the command line:

```
orca_blockf Fock.gbw Orbitals.gbw Output.gbw firstOrb lastOrb
```

Usage requires two `.gbw` files. One from which the Fock matrix will be reconstructed (Fock.gbw in the above) and another with the orbitals to be recanonicalized (Orbitals.gbw in the above). If the user doesn't provide an output `gbw` name, the program will overwrite the second argument `gbw` file.

This program is further described in section *Local Zero-Field Splitting*.

## 9.2.7 orca\_plot

`orca_plot` produces plot-ready files (such as `.plt` and `.cube`) for a range of orbital and density types computed with ORCA, enabling visualization with external 2D and 3D graphics software. It can be called in two ways:

- 1) within an ORCA input file via the `%plots` block. This approach allows you to generate graphical data (e.g., `.plt` or `.cube` files) automatically during the run. Details are provided in section *Orbital and Density Plots*.
- 2) interactively after a calculation, using the already generated `.gbw` file with following available parameters:

```
gbwfile      # name of gbw-file
-i           # interactive use of orca_plot
-m 256       # max. memory in MB (if needed)
```

If needed, the `-m`-option can be used to control the memory usage of the plotting job. `orca_plot` can be called in the terminal, for example, for the `gbw` file named `my.gbw`, as:

```
orca_plot my.gbw -i
```

This will list the available options:

```
1 - Enter type of plot
2 - Enter no of orbital to plot
3 - Enter operator of orbital (0=alpha,1=beta)
4 - Enter number of grid intervals
5 - Select output file format
6 - Plot CIS/TD-DFT difference densities
7 - Plot CIS/TD-DFT transition densities
8 - Set AO(=1) vs MO(=0) to plot
9 - List all available densities
10 - Perform Density Algebraic Operations

11 - Generate the plot
12 - exit this program
```

In the following, let us demonstrate the use of `orca_plot` for different plot types using the above option list.

## Perform Orbital Plots

Let us consider an ORCA input file for a single-point energy calculation on the pyridine molecule:

```
! RHF def2-SVP

*xyz 0 1
C      0.690940233      0.417992301     -1.170801378
C      0.690940233      1.616339301     -0.458357378
C      0.690940233      1.560238301      0.936438622
N      0.690940233      0.417992301      1.635468622
C      0.690940233     -0.724253699      0.936438622
C      0.690940233     -0.780354699     -0.458357378
H      0.690940233      0.417992301     -2.257043378
H      0.690940233      2.574997301     -0.967574378
H      0.690940233      2.478336301      1.521602622
H      0.690940233     -1.642351699      1.521602622
H      0.690940233     -1.739012699     -0.967574378
*
```

We want to plot the HOMO orbital. First, we need to determine its orbital number (20) from the ORBITAL ENERGIES section of the ORCA output file:

```
-----
ORBITAL ENERGIES
-----

NO    OCC      E (Eh)      E (eV)
 0    2.0000    -15.563726   -423.5105
...
Occupied Orbitals Manifold
...
 20    2.0000    -0.349834    -9.5195
...
Unoccupied Orbitals Manifold
...
 21    0.0000      0.111722      3.0401
...
```

Then let us specify appropriate choices for options 2, 3, 4, 5 and 8:

```
Enter a number: 2
Enter MO: 20

Enter a number: 3
Enter OP: 0

Enter a number: 4
Enter NGRID: 80

Enter a number: 5
File-Format is presently: 7
(7 - 3D Gaussian cube)

Enter a number: 8
Enter 0(MO) or 1(AO): 0
```

To generate the plot, choose 11. Then, exit `orca_plot` with choose 12 if no additional plots are needed. Otherwise, multiple plots can be requested sequentially in a single run.

```
11 - Generate the plot
Enter a number: 11 =>
```

(continues on next page)

(continued from previous page)

```

PlotType      ... MO-PLOT
MO/Operator   ... 20 0
Output file   ... pyridine_scf.mo20a.cube
Format        ... Grid3d/Cube
Resolution    ... 80 80 80

Calling PlotGrid3d with ATOM=A,B=0,0
Entering PlotGrid3d with Plottype =1
          *** PLOTTING FINISHED ***
Output file: pyridine_scf.mo20a.cube

```

This generates a .cube file that can be used to visualize the HOMO orbital with a visualization program. The picture of the HOMO orbital obtained using this cube file as input in Chimera is shown in Fig. 9.4.

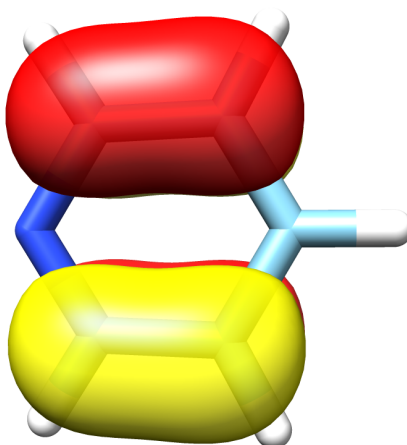


Fig. 9.4: Pyridine HOMO

## List of Density Plots

If a density instead of an orbital plot is required options 1 and 9 can be used to list the available densities.

For example option 1 will provide the following computed available densities for the above pyridine example.

```

Enter a number: 1
-----
Reading Over   2 Saved Densities                      ...
-----
Plot-Type is presently: 1
-----
Searching for Ground State Electron or Spin Densities: ...
-----
  1 - molecular orbitals
  2 - (scf) electron density          ..... (scfp      ⌞
  ↳      ) => AVAILABLE
  3 - (scf) spin density              ..... (scfr      ⌞
  ↳      ) - NOT AVAILABLE
  4 - natural orbitals

```

(continues on next page)

(continued from previous page)

5 -	corresponding orbitals		
6 -	atomic orbitals		
7 -	mdci electron density	.....	(mdcip <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
8 -	mdci spin density	.....	(mdcir <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
9 -	OO-RI-MP2 density	.....	(pmp2re <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
10 -	OO-RI-MP2 spin density	.....	(pmp2ur <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
11 -	MP2 relaxed density	.....	(pmp2re <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
12 -	MP2 unrelaxed density	.....	(pmp2ur <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
13 -	MP2 relaxed spin density	.....	(rmp2re <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
14 -	MP2 unrelaxed spin density	.....	(rmp2ur <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
15 -	LED dispersion interaction density	.....	(ded21 <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
16 -	Atom pair density		
17 -	Shielding Tensors		
18 -	Polarisability Tensor		
19 -	AutoCI relaxed density	.....	(autocipre <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
20 -	AutoCI unrelaxed density	.....	(autocipur <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
21 -	AutoCI relaxed spin density	.....	(autocirre <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		
22 -	AutoCI unrelaxed spin density	.....	(autocirur <span style="color: red;">_</span>
↩	) - NOT AVAILABLE		

In this case, only the SCF electron density (option 2) is available, and can thus be processed using `orca_plot` to prepare an input file for visualization software. Choose this option to generate the necessary `scfp` file:

```
Enter Type: 2
The default name of the density would be: pyridine_scf.scfp
Is this the one you want (y/n)?
```

The generated file will then appear under option 9 in the list of available density files.

```
-----
List of density names
-----
```

Index:	Name of Density
0:	pyridine_scf.scfp

Following the steps described for generating the cube file of the HOMO orbital, a file (cube, plt, etc.) necessary for visualizing the SCF electron density of pyridine can also be generated. When option 11 is selected, the following will be printed out in the terminal.

```
PlotType      ... DENSITY-PLOT
ElDens File   ... pyridine_scf.scfp
Output file    ... MyElDens
Format        ... Grid3d/Cube
Resolution    ... 80 80 80
```

```
Calling PlotGrid3d with ATOM-A,B=0,0
Entering PlotGrid3d with Plotttype =2
```

(continues on next page)

(continued from previous page)

```
*** PLOTTING FINISHED ***  
Output file: pyridine_scf.eldens.cube
```

The plot of this SCF electron density using the generated cube file as input to Chimera is as in Fig. 9.5.

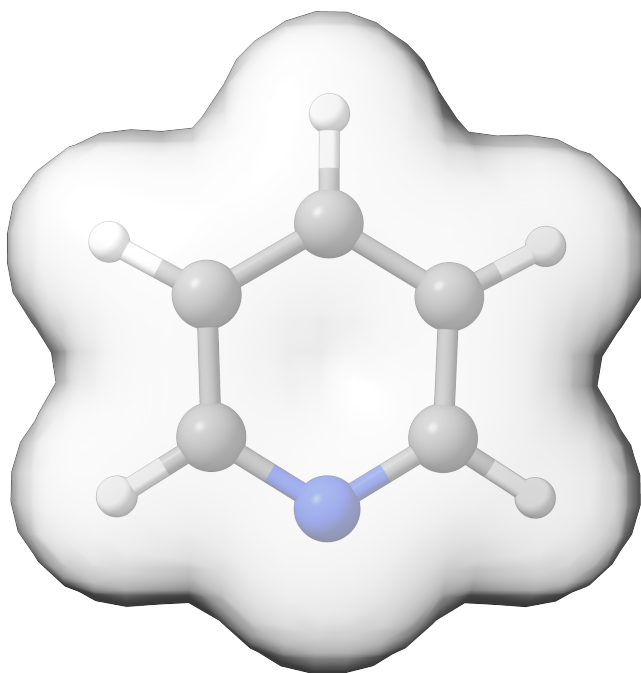


Fig. 9.5: Pyridine SCF Electron Density

### Perform Algebraic Operations

Starting from ORCA 6, one can perform simple algebraic operations with the computed densities. The presently available operations are listed in menu option 10:

```
Enter a number: 10
```

```
-----  
Available Algebraic Operations:  
-----
```

- 1 - Pair Densities Addition
- 2 - Pair Densities Subtraction

(continues on next page)

(continued from previous page)

- 3 - Pair Densities Multiplication
- 4 - Pair Densities Division
- 5 - Density Normalization
- 6 - Make Natural Transition Orbitals
- 7 - Make Natural Difference Orbitals
- 8 - Leave Section - Return to Main Menu

To perform such operations, the desired state or transition-state densities must be available in the `.densities` file. This can be achieved by requesting the relevant densities to be written to disk using the `!KeepDens` or `!KeepTransDensity` keywords in the input.

NOTE: Storing hundreds or thousands of such densities requires care, as they may occupy hundreds of gigabytes of disk space.

In addition to the SCF calculation on pyridine, let us now perform a canonical CCSD calculation with the following input:

```
! CCSD def2-SVP KeepDens

*xyz 0 1
C      0.690940233      0.417992301      -1.170801378
C      0.690940233      1.616339301      -0.458357378
C      0.690940233      1.560238301      0.936438622
N      0.690940233      0.417992301      1.635468622
C      0.690940233      -0.724253699      0.936438622
C      0.690940233      -0.780354699      -0.458357378
H      0.690940233      0.417992301      -2.257043378
H      0.690940233      2.574997301      -0.967574378
H      0.690940233      2.478336301      1.521602622
H      0.690940233      -1.642351699      1.521602622
H      0.690940233      -1.739012699      -0.967574378
*
```

Option 1 in the `orca_plot` menu will now provide both the scf and mdcf electron densities.

```
-----
Plot-Type is presently: 1
-----
Searching for Ground State Electron or Spin Densities:      ...
-----
  1 -   molecular orbitals
  2 -   (scf) electron density      ..... (scfp      ↵
↪   ) => AVAILABLE
  3 -   (scf) spin density      ..... (scfr      ↵
↪   ) - NOT AVAILABLE
  4 -   natural orbitals
  5 -   corresponding orbitals
  6 -   atomic orbitals
  7 -   mdcf electron density      ..... (mdcip      ↵
↪   ) => AVAILABLE
```

Let us compute correlation electron density by taking the difference of CCSD and HF electron densities. Then, we need to choose option 2 (Pair Densities Subtraction) of the Available Algebraic Operations menu:

```
-----
List of density names
-----

Index:                                     Name of Density
-----
  0:                                     pyridine_ccsd.scfp
```

(continues on next page)

(continued from previous page)

```

1:                                pyridine_ccsd.P0.tmp
2:                                pyridine_ccsd.mdcip

-----
Performing Algebraic Operations Over Densities: => SUBTRACTION
-----
Number of Densities to be Processed from the List => 2:
-----
Enter FileName for Density[ 0]: pyridine_ccsd.scfp
Provide a Scale Factor for Density[ 0] (Default => 1.00): 1
Enter FileName for Density[ 1]: pyridine_ccsd.mdcip
Provide a Scale Factor for Density[ 1] (Default => 1.00): 1
-----
INTERPRETTING EQUATION:
-----
1/sqrt(N) * [(1.00) * {pyridine_ccsd.scfp} - (1.00) * {pyridine_ccsd.mdcip}]
-----

PlotType      ... DENSITY-PLOT
ElDens File   ... pyridine_ccsd.scfp_minus_pyridine_ccsd.mdcip
Output file   ... pyridine_ccsd.scfp_minus_pyridine_ccsd.mdcip.cube
Format        ... Grid3d/Cube
Resolution    ... 80 80 80

```

The generated `pyridine_ccsd.scfp_minus_pyridine_ccsd.mdcip.cube` file can be directly visualized with a software like Chimera (see Fig. 9.6).

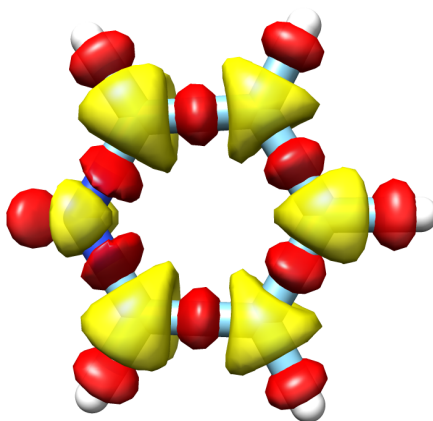


Fig. 9.6: Pyridine CCSD-HF Electron Density

NOTE: The generated density is stored in the Density Container so that it can be further processed or properly stored for future use. Now when option 9 is selected, the difference density will be seen among the available densities.

```

-----
List of density names
-----

```

Index:	Name of Density
--------	-----------------

(continues on next page)

(continued from previous page)

```

-----
0:                               pyridine_ccsd.scfp
1:                               pyridine_ccsd.P0.tmp
2:                               pyridine_ccsd.mdcip
3:                               pyridine_ccsd.scfp_minus_pyridine_ccsd.mdcip

```

## Plot Natural Transition Orbitals (NTOs) and Natural Difference Orbitals (NDOs)

`orca_plot` can also be used to generate Natural Transition Orbitals (NTOs) and Natural Difference Orbitals (NDOs) from any available state or transition density, regardless of the level of theory used. Let us now demonstrate this on the pyridine molecule for a state-averaged CASSCF(7,8) calculation. For this analysis, the relevant densities need to be stored to disk by including the `!KeepTransDensity` keyword in the input line. A corresponding ORCA input is as follows:

```

! def2-SVP KeepTransDensity

%casscf
  nel 8
  norb 7
  mult 1
  nroots 10
end

*xyz 0 1
C      0.690940233      0.417992301      -1.170801378
C      0.690940233      1.616339301      -0.458357378
C      0.690940233      1.560238301      0.936438622
N      0.690940233      0.417992301      1.635468622
C      0.690940233     -0.724253699      0.936438622
C      0.690940233     -0.780354699     -0.458357378
H      0.690940233      0.417992301     -2.257043378
H      0.690940233      2.574997301     -0.967574378
H      0.690940233      2.478336301      1.521602622
H      0.690940233     -1.642351699      1.521602622
H      0.690940233     -1.739012699     -0.967574378
*

```

When this calculation is run, in addition to the CASSCF electron density, all relevant CASSCF state and transition densities become available, as can be seen by choosing option 1 in the main `orca_plot` menu:

```

-----
Plot-Type is presently: 1
-----
Searching for Ground State Electron or Spin Densities:      ...
-----
  1 -   molecular orbitals
  2 -   (scf) electron density      ..... (scfp      ↵
↵   ) => AVAILABLE
  3 -   (scf) spin density      ..... (scfr      ↵
↵   ) - NOT AVAILABLE
  ...
-----
Searching for State or Transition State AO Electron Densities:      ...
-----
  23 -   CIS unrelaxed transition AO density      ..... (Tdens-CIS ↵
↵   ) - NOT AVAILABLE
  24 -   ROCIS unrelaxed transition AO density      ..... (Tdens-
↵ROCIS   ) - NOT AVAILABLE

```

(continues on next page)



(continued from previous page)

```

25 - CAS unrelaxed transition AO density          ..... (Tdens-CAS
↪      ) => AVAILABLE
...
-----
Searching for State or Transition State MO Electron Densities:    ...
-----
29 - CIS unrelaxed transition MO density          ..... (Tdens-
↪CISMO      ) - NOT AVAILABLE
30 - ROCIS unrelaxed transition MO density        ..... (Tdens-
↪ROCISMO    ) - NOT AVAILABLE
31 - CAS unrelaxed transition MO density          ..... (Tdens-
↪CASMO      ) => AVAILABLE
...

```

We can now proceed to generate the NTOs and NDOs that dominate the computed State 1.

```

ROOT   1:  E=   -246.3609192216 Eh   5.176 eV  41749.8 cm**-1
        0.82391 [   346]: 2212100
        0.06447 [   295]: 2112200

```

With options 6 =>NTOs of the Available Algebraic Operations menu, one finds AO transition density  $D_{01}$  file for NTOs as:

```

-----
Performing Algebraic Operations Over Densities: => MAKE_NTOS
-----
Enter FileName for Density[ 0]: Tdens-CAS-0-0-0-1
Provide a Scale Factor for Density[ 0] (Default => 1.00): 1
-----
NATURAL TRANSITION ORBITALS GENERATION:
-----
Warning: The one-electron matrix doesn't exist - is recalculated (SHARK)
Calculating the overlap matrix          ... done!
-----
NATURAL TRANSITION ORBITALS FOR STATE      1 1A
-----

STATE      1 1A  :  E=   0.190226 au      5.176 eV   41749.8 cm**-1

Threshold for printing occupation numbers 1.0000e-04

    0 : n=  1.28519446
    1 : n=  0.03997952
    2 : n=  0.01505089
    3 : n=  0.00336119

=> Natural Transition Orbitals (donor   ) were saved in Tdens-CAS-0-0-0-1.1-1A_nto-
↪donor.gbwn
=> Natural Transition Orbitals (acceptor) were saved in Tdens-CAS-0-0-0-1.1-1A_nto-
↪acceptor.gbwn
-----
Provide a Number of NTOs to plot (Default => 1): 1
-----
Reading Donor NTO-file      : Tdens-CAS-0-0-0-1.1-1A_nto-donor.gbwn
-----
Generating cube file for Donor NTO[0]:=> Tdens-CAS-0-0-0-1.1-1A_nto-donor.gbwn.0.
↪cube
-----
Reading Acceptor NTO-file   : Tdens-CAS-0-0-0-1.1-1A_nto-acceptor.gbwn

```

(continues on next page)

(continued from previous page)

```

-----
Generating cube file for Acceptor NTO[0]:=> Tdens-CAS-0-0-0-1.1-1A_nto-acceptor.
↪gbw.0.cube
Current-settings:

PlotType      ... MO-PLOT
MO/Operator   ... 0 0
Output file    ... Tdens-CAS-0-0-0-1.1-1A_nto-acceptor.gbw.0.cube
Format        ... Grid3d/Cube
Resolution    ... 80 80 80

```

With options 7 =>NDOs of the Available Algebraic Operations menu, one finds AO state density  $D_{00} - D_{11}$  file for NDOs as:

```

-----
Performing Algebraic Operations Over Densities: => MAKE_NDOS
-----
Enter FileName for Density[ 0]: Tdens-CAS-0-0-0-0
Provide a Scale Factor for Density[ 0] (Default => 1.00): 1
Enter FileName for Density[ 1]: Tdens-CAS-0-0-1-1
Provide a Scale Factor for Density[ 1] (Default => 1.00): 1
-----
NATURAL DIFFERENCE ORBITALS GENERATION:
-----
Warning: The one-electron matrix doesn't exist - is recalculated (SHARK)
Calculating the overlap matrix      ... done!
-----
NATURAL DIFFERENCE ORBITALS FOR STATE      1 1A
-----

STATE      1 1A  : E=   0.190226 au      5.176 eV      41749.8 cm**-1

Threshold for printing occupation numbers 1.0000e-04

    0 : n=  0.49881235
    1 : n=  0.08492368
    2 : n=  0.06708141
    3 : n=  0.01254920
    4 : n=  0.00667407
    5 : n=  0.00006162

=> Natural Difference Orbitals (donor   ) were saved in Tdens-CAS-0-0-0-0-Tdens-
↪CAS-0-0-1-1.1-1A_ndo-donor.gbw
=> Natural Difference Orbitals (acceptor) were saved in Tdens-CAS-0-0-0-0-Tdens-
↪CAS-0-0-1-1.1-1A_ndo-acceptor.gbw
-----
Provide a Number of NDOs to plot (Default => 1): 1
-----
Reading Donor NDO-file      : Tdens-CAS-0-0-0-0-Tdens-CAS-0-0-1-1.1-1A_ndo-donor.gbw
-----
Generating cube file for Donor NDO[0]:=> Tdens-CAS-0-0-0-0-Tdens-CAS-0-0-1-1.1-1A_
↪ndo-donor.gbw.0.cube
-----
Reading Acceptor NDO-file    : Tdens-CAS-0-0-0-0-Tdens-CAS-0-0-1-1.1-1A_ndo-
↪acceptor.gbw
-----
Generating cube file for Acceptor NDO[0]:=> Tdens-CAS-0-0-0-0-Tdens-CAS-0-0-1-1.1-
↪1A_ndo-acceptor.gbw.0.cube
Current-settings:

```

(continues on next page)

(continued from previous page)

```

PlotType      ... MO-PLOT
MO/Operator   ... 0 0
Output file   ... Tdens-CAS-0-0-0-0-Tdens-CAS-0-0-1-1-1A_ndo-acceptor.gbw.0.cube
Format        ... Grid3d/Cube
Resolution    ... 80 80 80

```

Using the resulting cube file, one can readily visualize the corresponding Donor and Acceptor NTOs and NDOs orbital pairs as in which can be readily visualized in Fig. 9.7.

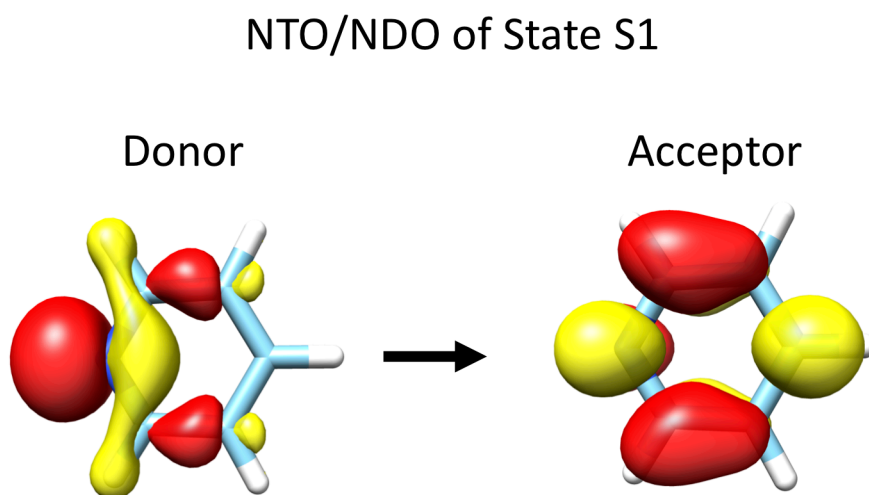


Fig. 9.7: Pyridine SA-CASSCF(8,7) NTO/NDO donor/acceptor orbital pairs for State 1

NOTE: It is both beneficial and more accurate to always process AO-basis densities. In particular, it is neither possible nor correct to process reduced MO-basis densities.

### Perform Electrostatic Potential Plots

Starting with ORCA 6.1, the electrostatic potential (ESP) can be generated in `orca_plot` for any state density available in the density container.

After performing an electronic structure calculation (e.g., using *input*), proceed as follows in `orca_plot`:

First, choose 1 - Enter type of plot, then select 43 - Electrostatic potential.

```

41 -   ROCIS QDPT unrelaxed transition AO density      ..... (Tdens-
->ROCISQDSOC      ) - NOT AVAILABLE
42 -   LFT QDPT unrelaxed transition AO density      ..... (Tdens-
->LFTQDSOC      ) - NOT AVAILABLE
-----
43 -   Electrostatic Potential

```

You will be prompted to specify the name of the state density for which the electrostatic potential is to be computed.

```

Enter Type: 43
-----
List of density names

```

(continues on next page)

(continued from previous page)

```
-----  
Index:                                     Name of Density  
-----  
    0:                                     pyridine_scf.scfp  
  
Enter Name for an STATE Density: pyridine_scf.scfp
```

Finally, select option 11 - Generate the plot to compute the electrostatic potential and generate the file `pyridine_scf.scfp.esp.cube`.

The resulting `.cube` file contains the electrostatic potential data and may be visualized directly or used to color a density plot within any compatible graphical user interface tool.

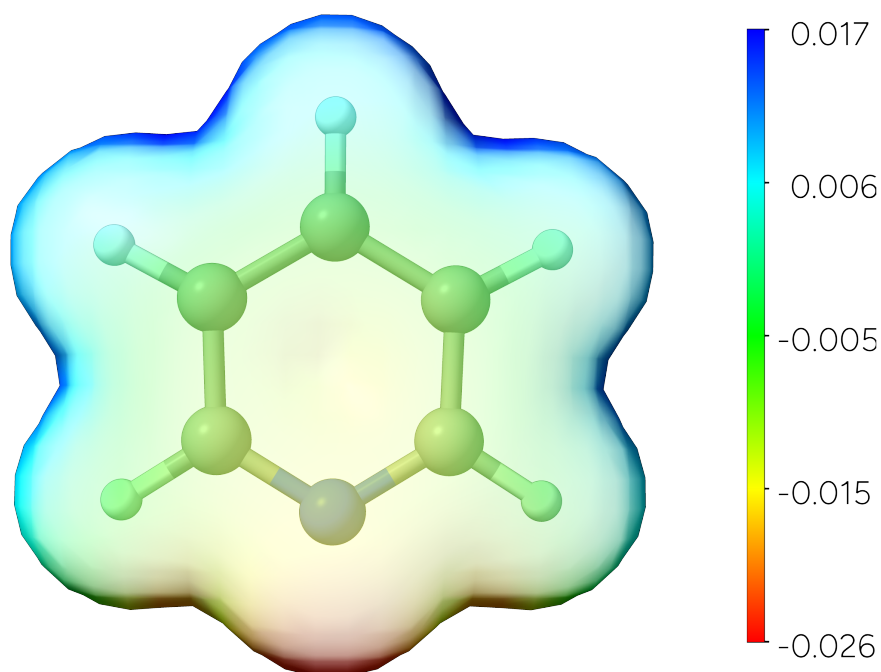


Fig. 9.8: Pyridine SCF Electron Density using ESP colouring

### Additional TIPS regarding Density Plots

- 1) As an alternative to menu option 9, one can directly check which densities are available in the calculation directory by reading out the list of all densities contained in the `.densities` file:

```
orca_plot mymol.densities
```

- 2) `orca_plot` can also be run in parallel mode for computationally expensive plots:

```
mpirun -np 4 /orca_path/orca_plot_mpi MyGBWFile.gbw -i MyGBWFile
```

- 3) It is possible to use `orca_plot` to create difference densities between the ground and excited states from CIS or TD-DFT calculations directly from the `.cis` calculation information. This is implemented as an extra interactive menu point that is (hopefully) self-explanatory. Starting from ORCA 6 one can use the Algebraic Operations utility menu to carry out these operations directly from the generated State and Transition Densities.
- 4) It is also possible to use `orca_plot` to create difference densities between the ground and excited states from CIS or TD-DFT calculations using the `.cis` file. This functionality is implemented as an additional interactive menu option, which is (hopefully) self-explanatory. Starting from ORCA 6, these densities can also be obtained through the Algebraic Operations menu, processing directly the generated state and transition densities.

### 9.2.8 orca\_2mkl: Old Molekel as well as Molden inputs

The `orca_2mkl` utility converts `.gbw` files into `.mkl` ASCII files. The `.mkl` file contains essentially the same information as the `.gbw` file, excluding ORCA-specific internal flags. The ASCII format makes it suitable for external use—for example, it can be read by Molekel for visualization or accessed by custom post-processing scripts. As such, `.mkl` files provide a convenient starting point for developing interfaces with third-party tools. `orca_2mkl` can be simply invoked as:

```
orca_2mkl BaseName
    (will produce BaseName.mkl from BaseName.gbw)
orca_2mkl BaseName -molden
    (writes a file in molden format)
orca_2mkl BaseName -mkl
    (writes a file in MKL format)
```

Any `gbw`-like file, containing such as QRO, UNO, UCO, etc. orbitals, can be converted to the `mkl` and `Molden` formats using `orca_2mkl`:

```
orca_2mkl gbw_type_file.extension mkl_file.extension -mkl -anyorbs
# or
orca_2mkl gbw_type_file.extension molden_file.extension -molden -anyorbs
```

`orca_2mkl` can also be run in reverse to generate `.gbw` type files from `.mkl` files:

```
orca_2mkl BaseName -gbw
    (will produce BaseName.gbw out of BaseName.mkl)
```

This functionality allows users to import molecular orbitals from external sources into ORCA. Although this feature is rarely used and offers limited capabilities, it is included here for users who wish to experiment. An example of this usage is demonstrated in the CASSCF tutorial accompanying the ORCA manual.

Finally, some third-party tools are available for converting the wavefunction files of other quantum chemistry programs from and to the MKL format, e.g. MOKIT (<https://gitlab.com/jxzou/mokit>). Combined with `orca_2mkl`, this provides a convenient way to use ORCA-generated molecular orbitals in other programs, or the other way around. One example application would be to use ORCA's TRAH method to converge a difficult SCF problem, and pass the converged orbitals to another program that does not have the TRAH converger. Another example is to read in the

orbitals of a difficult system from another program, and compute its high-level single point energy using ORCA using a functional or method that is not supported by the other program, without having to converge the wavefunction from the PModel guess. Finally, MOKIT can convert the MKL format to the PySCF format, which allows one to manipulate ORCA molecular orbitals in a Python environment.

### 9.2.9 orca\_2aim

This utility program reads a .gbw file and creates a .wfn and .wfx file that can be used for topological analysis of the electron density by other programs. This works for open-shell and closed-shell wave functions. It can be invoked by including ! AIM in the simple input line of the ORCA input file, or typing in the terminal as follows:

```
orca_2aim BaseName
      (will produce BaseName.wfn and BaseName.wfx from BaseName.gbw)
```

### 9.2.10 orca\_vpot

This orca\_vpot program calculates the electrostatic potential at a given set of user defined points. It can be used with either an input file or in interactive mode. It needs four arguments:

```
orca_vpot MyJob.gbw MyJob.scfp MyJob.vpot.xyz MyJob.vpot.out
```

*First:* The gbw file containing the correct geometry and basis set

*Second:* The desired density matrix in this basis (perhaps use the KeepDens keyword)

*Third:* an ASCII file with the target positions in AU, e.g.

```
6          (number_of_points)
5.0 0.0 0.0 (XYZ coordinates)
-5.0 0.0 0.0
0.0 5.0 0.0
0.0-5.0 0.0
0.0 0.0 5.0
0.0 0.0 -5.0
```

*Fourth:* The target file which will then contain the electrostatic potential, e.g.

```
6          (number of points)
VX1  VY1  VZ1 (potential for first point)
VX2  VY2  VZ2 (potential for second point)
etc.
```

It is straightforward to read this file and use the potential for intended purpose.

orca\_vpot can be called in parallel mode as follows:

```
mpirun -np 4 /full_path/orca_vpot_mpi MyJob.gbw MyJob.scfp MyJob.pot.xyz MyJob.pot.
↪out
```

It can be used to check available densities in MyJob.densities as:

```
orca_vpot MyJob.densities
```

If the base names of the .gbw and .densities files do not match, the base name of the .densities file must be passed as the fifth argument:

```
orca_vpot MyJob.gbw OtherJob.scfp MyJob.pot.xyz MyJob.pot.out OtherJob
```

A call to orca\_vpot without any arguments will display a help message.

### 9.2.11 orca\_euler

The `orca_euler` utility program calculates the relative orientation between calculated hyperfine coupling (HFC) or nuclear quadrupole coupling (NQC) tensors and a reference tensor, typically the g-tensor or the D-tensor. The `orca_euler` program is invoked automatically at the end of an ORCA job if both the HFC/NQC and g-/D-tensor are calculated within the same run. Alternatively, it can be executed as a standalone program, provided that the `.property.txt` file from a previous NQC/HFC- and D- or g-tensor calculation is available.

The orientation between tensors is expressed as a 3×3 rotation matrix  $R$ , parameterized by the three Euler angles:  $\alpha$ ,  $\beta$  and  $\gamma$ . These angles define the relative orientation of tensor A with respect to tensor B through three successive rotations around different axes to align A with B, i.e., in the commonly used z-y-z convention:

- **Rotate  $A_{xyz}$  counterclockwise around its  $z$  axis by  $\alpha$  to obtain  $A_{x'y'z'}$ .**
- **Rotate  $A_{x'y'z'}$  counterclockwise around its  $y'$  axis by  $\beta$  to obtain  $A_{x''y''z''}$ .**
- **Rotate  $A_{x''y''z''}$  counterclockwise around its  $z''$  axis by  $\gamma$  to align it with B.**

The options of the `orca_euler` program are as follows:

```
orca_euler prop-file options

file      = basename of an ORCA .property.txt file

options
-refg/-refD:      Reference tensor (g-tensor or D-tensor, default is -refg)
-conv zyz/-conv zxz: Euler rotation convention (default is zyz)
-order:          Ordering of the reference tensor (x, y, z) with respect to
                  ORCA output (min, mid, max)
-plotA:          plot the HFC-tensors
-plotQ:          plot the NQC-tensors
-detail:         print detailed information
```

NOTE:

- By default the D-tensor is used as reference tensor only if  $S > \frac{1}{2}$  and if  $D > 0.3 \text{ cm}^{-1}$ . In all other cases, the g-tensor is used as reference tensor. The user can manually select the reference tensor (if available in the `.property.txt` file) by using `-refg` or `-refD`.
- By default, the Euler rotation is performed using the z-y-z convention. The z-x-z convention can be selected manually by using the option `-conv zxz`.
- By default, the axes of the g- or D-tensor are assigned based on the magnitude of their components:  $g_{\min} \rightarrow g_x$ ,  $g_{\text{mid}} \rightarrow g_y$ ,  $g_{\max} \rightarrow g_z$  (similarly for D). This ordering can be modified manually when running the standalone program, using the `-order` option as shown below:

-order 3 2 1:	min → z
	mid → y
	max → x
-order 1 -2 3:	min → x
	mid → y (flipped in the orientation)
	max → z

- Files in `.xyz` format can be generated to plot nuclear hyperfine and quadrupole coupling tensors using the `orca_euler` program with the option `-plotA` or `-plotQ`. For example, the HFC tensor of atom 3 (counting starts at zero) is saved as `prop-file.3.A.xyz`, and the corresponding NQC tensor as `prop-file`.

3.Q.xyz. These xyz files include four fictitious atoms (He, Ne, Ar, Kr), where the directions of the tensor axes are defined as vectors: x-axis from He to Ne; y-axis from He to Ar; and z-axis from He to Kr.

- To print the actual definition of the rotation matrix and further information on the relative orientation, use the `-detail` option.

## 9.2.12 orca\_exportbasis

The `orca_exportbasis` utility program prints out the basis sets used by ORCA. It requires the name of the basis set as specified in the ORCA input line. Optional arguments include an ECP basis set, a list of specific atoms, and the name of an output file. The output is written in ASCII format, which can be viewed and manually edited. Users can choose to export the basis sets in either ORCA format, which can be copied directly into the input file, or in GAMESS-US format, which can be used via the `%basis` block as externally specified basis.

**NOTE:** Basis set names containing special characters may require a pair of quotation marks (” or ‘) to be correctly recognized.

The usage and options of `orca_exportbasis` are as follows:

```
USAGE: orca_exportbasis keywords options
```

```
-b, --basis : name of basis set
              def2-svp
              'def2-tzvp(-f)' - string to be passed with literals
```

```
EXAMPLE: orca_exportbasis -b svp
```

Additional Options:

```
-e, --ecp : ecp basis
            sdd
            default - ECP-part of basis (if present)

-f, --format : output format
              ORCA - to be read via %basis NewGTO
              GAMESS-US - to be read as %basis GTOName 'mybasis.bas'
              default - ORCA

-a, --atoms : list of elements
              Cu - single element
              Ga Ge As Se - list of elements separated by blanks
              default - whole periodic table is printed

-o, --outfile: name of outputfile
              mybasis.bas
              default - derived name
```

```
EXAMPLE: orca_exportbasis -b svp -e sdd -a Ag -f GAMESS-US -o mybasis.bas
```

The output file stored in GAMESS-US format can be called in the `%basis` block of the next ORCA calculation - see *Reading Basis Sets from a File*.

```
%basis
GTOName      "mybasis.bas"
GTOAuxJName  "myauxjbasis.bas"
GTOAuxJKName "myauxjkbasis.bas"
GTOAuxCName  "myauxjcbasis.bas"
end
```



### 9.2.13 orca\_eca

The `orca_eca` utility program uses the calculated exchange coupling constants to compute relative energies of all possible spin states by diagonalizing the spin Hamiltonian. The absolute and relative energies of the resulting spin states are printed in the `*.en` and `*.en0` files, respectively. In addition, the on-site spin expectation values are written to a `*.sp` file. The following example calculates the spin ladder for a system with exchange coupling constant of  $-152.48 \text{ cm}^{-1}$  between Mn(III) and Mn(IV).

```
%sim
ms_bs 0.5 # Arbitrary spin state
end

# specification of spin centers
$spins 2
1 2.0 # Spin on first manganese
2 1.5 # Spin on second manganese

# Exchange coupling constant ( $H = -2J S_1 S_2$ )
$ecc 1
1 -152.48

$aiso_bs 2 # A false segment just to print the *.sp file
1 0.00
2 0.00
```

### 9.2.14 orca\_pnmr

The `orca_pnmr` program calculates the paramagnetic contribution to the NMR shielding tensor using the EPR  $g$ ,  $A$ , and  $D$  tensors (see Section [Paramagnetic NMR Shielding Tensors](#) for theoretical background). It is a standalone program that can be executed from the command line after the main ORCA calculation has finished. Alternatively, it can read user-provided  $g/A/D$  tensors from an input file using the option `-i`. Note that `orca_pnmr` expects  $g$  and  $A$  tensors to follow the conventions described in Section [Cartesian Index Conventions for EPR and NMR Tensors](#).

The usage and options of `orca_pnmr` are as follows:

```
USAGE:      orca_pnmr BaseName [-i] [-v]

OPTIONS:   -i : read from the input file "BaseName.pnmr.inp"
           -v : print more output (Z matrices)
```

When run without any options, `orca_pnmr` attempts to extract the  $g$ ,  $D$ , and  $A$  tensors from the file `BaseName.property.txt` and compute the paramagnetic shieldings at 298 K. This functionality only works if the tensors were computed using the EPRNMR module. A more flexible and controlled approach is to manually edit the `BaseName.pnmr.inp` (see below for its content) and then run `orca_pnmr` with the option `-i`.

```
3      # Spin multiplicity (2S+1)
298    # Temperature range minimum [K]
300    # Temperature range maximum [K]
1      # Temperature step [K]
1      # Have g-tensor? 0 or 1
2.004689 0.000000 -0.000000 # Cartesian g-tensor
0.000000 2.004689 0.000000 # (if available)
0.000000 0.000000 2.002123
1      # Have D-tensor? 0 or 1
-0.514341 -0.000000 -0.000000 # Cartesian D-tensor [cm-1]
-0.000000 -0.514341 -0.000000 # (if available)
-0.000000 -0.000000 1.028682
2      # Number of A-tensors
00     # First nucleus (index,element)
-72.358765 # Prefactor [MHz]
```

(continues on next page)

(continued from previous page)

```

-78.989514  0.000000  -0.000000 # Cartesian A-tensor [MHz]
  0.000000 -78.989514  -0.000000
-0.000000  -0.000000  53.478581
10 # Next nucleus (index,element)
-72.358765 # Prefactor [MHz]
-78.989516  0.000000  -0.000000 # Cartesian A-tensor [MHz]
  0.000000 -78.989516  -0.000000
-0.000000  -0.000000  53.478580
# ... Further nuclei

```

- The  $D$  and  $g$  tensors are optional. If they are not provided, `orca_pnmr` assumes an isotropic free-electron  $g$  value and sets  $D = 0$ .
- The  $A$  tensors are required. If no  $A$  tensor is provided for a given nucleus, the paramagnetic NMR shielding cannot be computed for that nucleus.

## 9.2.15 orca\_lft

Starting from ORCA 5.0, ORCA features a standalone multiplet program called `orca_lft`.

- `orca_lft` is dedicated to experimental spectroscopists.
- It is able to run an arbitrary number of spectra simulations with emphasis on X-ray spectroscopies.

In this section we briefly review the main functionalities of `orca_lft`. For a more detail description and examples discussion, please refer to the `orca_lft` tutorial.

1. The goal is to be able to compute various spectroscopic properties of a given LFT center (ion) if one can manually pass the information of 1 and 2 electron integrals in the form of e.g the diagonal elements of the LFT matrix (LFT orbital energies) and the Slater-Condon parameters of a given LFT problem.
2. This will allow the experimental spectroscopist to perform a massive amount of spectra simulations during the actual running experiments

Any LFT problem can be parametrized in terms 1-electron  $H_{LF}$  matrix elements and the Slater-Condon 2-electron integrals  $F0$ ,  $F2$ ,  $F4$ , (or the Racah parameters  $A$ ,  $B$ ,  $C$ , of the d-shell). (Figure: Fig. 9.9)

$$\hat{H}_{LF} = \overbrace{\sum_i \hat{v}_{LF}(i)}^{1\text{-electron integrals}} + \overbrace{\sum_{i < j} \hat{G}_{LF}(i, j)}^{2\text{-electron integrals}}$$

Fig. 9.9: Definition of an LFT problem in terms of 1-electron energies and Slater Condon parameters (SCPs)

In practice we need to know:

1. the Slater Condon parameters of a given LFT problem
2. the  $H_{LF}$  matrix elements or the relation of them (ligand field splitting, 10Dq, AOM model)

The design workflow of `orca_lft` is the following:

- Solve the General CI problem on a User-specified LFT problem: Type of Ion, Number of Electrons, Involved Shells, Involved Multiplicities
- Compute All possible Non Relativistic States/Multiplicity
- Compute the Transition Densities on the CSFs basis
- Compute the Needed transition Moments in the given LFT basis (i.e. 2p3d)

- Compute various properties with emphasis to X-ray spectroscopy (ABS, XAS, XES, RIXS) at the Non Relativistic Limit
- Compute the respective Relativistically corrected States on the Quasi Degenerate Perturbation Theory basis
- Provide access to various relativistically corrected properies (ABS, XAS, XES, RIXS, MCD, XMCD, GTensors, Densors, Hyperfine Couplings, Electric Field Gradients)

orca\_lft requires its own input. By simply executing it from the terminal

```
orca_lft
```

one gets printings for the usage:

```
*****
Simulate or Fit Spectra
*****

=====
Usage: orca_lft BaseName.lft.inp [options]
=====

-----
->-----
[Options]:
-----
->-----
-sim                Simulate Spectra
-fit                Fit Spectra (This is not yet available!)
-----
->-----

*****
Generate Initial Input
*****

=====
Usage: orca_lft BaseName [options]
=====
```

various Run Options:

```
-----
->-----
[Options]:
-----
->-----
-p_case            Requests  p-shell case
-d_case            Requests  d-shell case
-f_case            Requests  f-shell case
-sp_case           Requests  sp-shell case
-ps_case           Requests  sp-shell case
-sd_case           Requests  sd-shell case
-ds_case           Requests  ds-shell case
-sf_case           Requests  sf-shell case
-fs_case           Requests  fs-shell case
-pd_case           Requests  pd-shell case
-dp_case           Requests  dp-shell case
-pf_case           Requests  pf-shell case
-fp_case           Requests  fp-shell case
-df_case           Requests  pf-shell case
-fd_case           Requests  fp-shell case
-spd_case          Requests  spd-shell case
```

(continues on next page)

(continued from previous page)

```

-spf_case           Requests spf-shell case
-sdf_case           Requests sdf-shell case
-pdf_case           Requests pdf-shell case
-soc                Requests Input with SOC Constants
-atnoN              Sets Atomic Number N
-----
↳ -----
-Special Cases for known Elements. LFT Parameters are filled from an internal_
↳ AILFT NEVPT2 database
-----
--Supported Oxidation States--
↳ -----
Presently Default Oxidation States are supported except for Fe:
Main Elements           -> 0
TM Elements (Fe(26))    -> I,II,III
TM Elements (All other) -> II
Lanthanide/Actinide Elements -> III
-----
--Valence Cases--
↳ -----
-atnoN -2s2p_case      Requests 2s2p-shell case for element with atomic_
↳ number N (4-9)
-atnoN -3s3p_case      Requests 3s3p-shell case for element with atomic_
↳ number N (12-18)
-atnoN -4s4p_case      Requests 4s4p-shell case for element with atomic_
↳ number N (20,31-36)
-atnoN -4s4p_case      Requests 5s5p-shell case for element with atomic_
↳ number N (38,49-54)
-atnoN -3d4s_case      Requests 3d4s-shell case for element with atomic_
↳ number N (22-29)
-atnoN -4d5s_case      Requests 4d5s-shell case for element with atomic_
↳ number N (40-47)
-atnoN -5d6s_case      Requests 4d6s-shell case for element with atomic_
↳ number N (72-79)
-atnoN -4f5d_case      Requests 4d5d-shell case for element with atomic_
↳ number N (59-70)
-atnoN -5f6d_case      Requests 5d6d-shell case for element with atomic_
↳ number N (91-102)
-----
--Core-Valence Cases--
↳ -----
-atnoN -1s3d_case      Requests 1s3d-shell case for element with atomic_
↳ number N (22-29)
-atnoN -2p3d_case      Requests 2p3d-shell case for element with atomic_
↳ number N (22-29)
-atnoN -3p3d_case      Requests 3p3d-shell case for element with atomic_
↳ number N (22-29)
-----
--Core-Valence XES Cases--
↳ -----
-atnoN -1s2p3d_case    Requests 1s2p3d-shell case for element with atomic_
↳ number N (22-29)
-atnoN -1s3p3d_case    Requests 1s3p3d-shell case for element with atomic_
↳ number N (22-29)
-----
↳ -----

```

and various Spectra Simulation Options:

```

*****
Spectra Simulation Options for the BaseName.lft.inp:
*****
-----

```

(continues on next page)