Running ORCA successively on both of these input files in the same directory will yield an output that contains the zero-point vibrational corrections to the shielding tensors for each atom. For Atom 0, which is the carbon in methanol, it looks like this:

```
Vibrationally averaged isotropic shieldings
Atom 0 :
mode <q>
                <q2> dS/dQ d2S/dQ2
  0 -0.000017 0.202578 -0.000089 -5.922644
  1 -0.034052 0.057707 8.269988 -5.666515
  2 -0.036827 0.055687 5.667278 -13.843941
  3 0.000002 0.051446 0.000073 -7.353936
  4 0.027471 0.043993 0.423409 -6.207061
  5 -0.009357 0.040649 -12.736464 3.762324
  6 \quad -0.000001 \quad 0.040278 \quad -0.001621 \quad -2.224536
  7 0.001277 0.039898 -1.266298 -3.916647
  8 -0.031609 0.020149 51.647411 -21.635780
  9 -0.000021 0.019859 0.035760 -61.239749
 10 -0.010397 0.019376 18.573156 -50.591165
 11 -0.026641 0.015808 -8.871055 -6.654795
zpv correction to isotropic shift: -4.840215 ppm
```

So the absolute shielding constant of carbon in methanol needs to be corrected by -4.8 ppm due to zero-point vibration. From the mean and mean square displacements and the first and second derivatives of the shieldings with respect to the normal modes, one can also identify degrees of freedom which give rise to larger contributions of the vibrational correction.

A similar input for the HH spin-spin coupling constants would look like this:

```
!TPSS pcJ-2 Autoaux ExtremeSCF NMR
%geom inhess read
inhessname "vpt2_methanol_FF.hess"
end
%maxcore 4096
%vpt2
 VPT2
           on
 AvgProp
           JCOUPLING
 AnharmDisp
           0.05
 HessianCutoff 1e-12
end
%met.hod
           1e-12
 7. Tol
end
%eprnmr
 Tol
           1e-10
end
* xyz 0 1
```

```
H 0.66732439683790 0.98510769198508 0.00001819506998

H -1.45583606337199 -0.88374271593276 0.00000595999622

H -1.49206267729630 0.64725244577978 0.89143349761200

H -1.49208273899904 0.64724452288014 -0.89144277697426

*

*eprnmr
Nuclei = all H {ssfc}
end
```

As mentioned above, the exact same procedure is also available for A-tensors. Here is an example for the NH_2 radical with the VPT2 input file vpt2 NH2 FF.inp:

```
!UKS BP86 def2-svp def2/J ExtremeSCF defgrid3
%vpt2
  VPT2
                 On
end
%method
                1e-12
  Z_Tol
end
* xyz 0 2
0.00000000000000
  -0.22855980523269
                  1.00639225931822
                                  0.00000000000000
Н
```

and the input file - something like $vpt2_NH2_A.inp$ - for the level of theory that will be used in the A-tensor computation:

```
!UKS BP86 def2-SVP TightSCF
%geom inhess read
inhessname "vpt2_NH2_FF.hess"
end
%vpt2
                On
  VPT2
                Atensor
  AvgProp
end
*xyz 0 2
 Η
%eprnmr
Nuclei = all N { aiso, adip }
Nuclei = all H { aiso, adip }
```

and similarly for the g-tensor if Atensor is replaced by Gtensor in the %vpt2 block (the %eprnmr block can be omitted then).

Note that a convenient way to compute vibrational corrections is the usage of a compound script. With an input file called $\mathtt{NH2.inp}$:

and the corresponding compound script NH2.cmp:

```
New_Step
!UHF def2-SVP VeryTightSCF
%vpt2
  VPT2
end
%met.hod
Z_Tol
               1e-12
end
* xyz 0 2
Step_End
New_Step
!UHF def2-SVP VeryTightSCF
%geom inhess read
inhessname "NH2_Compound_1.hess"
end
%vpt2
            On
Atensor
  VPT2
  AvgProp
end
*xyz 0 2
%eprnmr
Nuclei = all N { aiso, adip }
Nuclei = all H { aiso, adip }
end
Step_End
END
```

a similar result can be obtained in one calculation.



Make sure the correct hessian file names are given and the input files MUST not contain a compound block. You can also rerun the VPT2 analysis using orca_vpt2 directly. If you have an anharmonic force field cal-

culation named myjob_ff and a property derivative calculation named myjob_prop just call orca_vpt
myjob_ff.vpt2 myjob_prop.vpt2.

5.20 Electrical Properties - Electric Moments and Polarizabilities

A few basic electric properties can be calculated in ORCA although this has never been a focal point of development. The properties can be accessed straightforwardly through the <code>%elprop</code> block:

```
! B3LYP DEF2-SVP TightSCF
%elprop Dipole
                  true # dipole moment
       Quadrupole true # quadrupole moment
                    true # dipole-dipole polarizability
                       1 # equivalent to true (for backward compatibility)
                         # Note: the flags "polar 2" and "polar 3" for_
⇔seminumeric
                         # and fully numeric polarizabilities are not supported
                         # anymore! For numerical polarizability calculations
                         # please use the respective compound scripts
                    true # the dip/dip/dip hyperpolarizability
       PolarVelocity true # polarizability w.r.t. velocity perturbations
       PolarDipQuad true # dipole-quadrupole polarizability
       PolarQuadQuad true # quadrupole-quadrupole polarizability
* int 0 1
  C 0 0 0 0
  H 1 0 0 1.09 109.4712
  H 1 2 0 1.09 109.4712
  H 1 2 3 1.09 109.4712 120
  H 1 2 3 1.09 109.4712 240
```

The polarizability (dipole-dipole, dipole-quadrupole, quadrupole-quadrupole) is calculated analytically through solution of the coupled-perturbed (CP-)SCF equations for HF and DFT runs (see *CP-SCF Options*) and through the CP-CASSCF equations for CASSCF runs (see *CASSCF Linear Response*). Analytic polarizabilities are also available for CCSD/CCSD(T) (see *AUTOCI Response Properties via Analytic Derivatives*), RI-MP2 and DLPNO-MP2, as well as double-hybrid DFT methods. For canonical MP2 one can use AUTOCI for analytic calculations (see *AUTOCI Response Properties via Analytic Derivatives*) or differentiate the analytical dipole moment calculated with relaxed densities. For other correlation methods only a fully numeric approach is possible.

```
0.00000170 -0.00000346
                                 12.868107945
diagonalized tensor:
                 12.861067290
   12.851869269
                                 12.868107945
Orientation:
    0.000017958
                                 -0.000050696
    -0.000004890
                   0.000053560
                                  0.999999999
Isotropic polarizability: 12.86035
```

As expected the polarizability tensor is isotropic.

Dipole-quadrupole polarizability tensors are printed as a list of 18 different components, with the first index running over x,y,z and the second index running over xx,yy,zz,xy,xz,yz. This is known as the "pure Cartesian" version of the tensor, although other conventions may exist in the literature that differ from the ORCA values by a constant factor.

```
STATIC POLARIZABILITY TENSOR (Dipole/Quadrupole)
Method : SCF
Type of density : Electron Density
Type of derivative : Electric Field (Direction=X)
Multiplicity : 1
Irrep
Relativity type
                    :
Basis
The raw cartesian tensor (atomic units):
 X- X X : 11.577165985
X- Y Y : -5.795339382
 X- Z Z :
              -5.797320742
 X- X Y: 0.001285565
X- X Z: 0.000000155
 X- Y Z : -0.000000077
Y- X X : 0.001386387
Y- Y Y : 8.200445841
  Y- Z Z :
               -8.198375727
  Y- X Y: -5.794687548
  Y- X Z :
               0.000000228
  Y- Y Z :
               -0.000000121
  Z- X X: -0.00000151
 Z- Y Y :
Z- Z Z :
                0.000000627
                -0.000000812
  Z- X Y :
Z- X Z :
                -0.000000312
               -5.798359323
  Z- Y Z :
               -8.205110537
```

After this, the "traceless" version of the tensor is printed, which is usually denoted by $A_{x,xx}, A_{x,xy}$ etc. in the literature [682, 683, 684]. This is the preferred format for reporting dipole-quadrupole polarizability tensors. Certain references use the opposite sign convention than reported here, but generally the conventions of traceless polarizability tensors are more unified than those of pure Cartesian polarizability tensors.

```
STATIC TRACELESS POLARIZABILITY TENSOR (Dipole/Quadrupole)
Method : SCF
Type of density : Electron Density
Type of derivative : Electric Field (Direction=X)
                                                                               (continues on next page)
```

```
Multiplicity : 1
Irrep
                  : 0
Relativity type
Basis
                  : AO
The raw cartesian tensor (atomic units):
 X- X X: 17.373496046
 X- Y Y :
             -8.685262003
 X-YY: -8.685262003
 X-ZZ: -8.688234043
             0.001928347
0.000000232
 X- X Y :
 X- X Z :
            -0.000000116
 X- Y Z :
 Y- X X :
              0.000351329
            12.298940512
 Y- Y Y :
           -12.299291841
 Y- Z Z :
             -8.692031322
 Y- X Y :
 Y- X Z :
              0.000000342
 Y- Y Z :
            -0.000000181
 Z- X X: -0.000000058
 Z- Y Y :
             0.000001109
 Z- Z Z :
            -0.000001050
 Z- X Y : -0.000000468
 Z- X Z :
             -8.697538984
 Z- Y Z : -12.307665806
```

The quadrupole-quadrupole polarizability tensor is similarly printed in both the pure Cartesian and traceless forms. Again, the traceless form (usually denoted as $C_{xx,xx}$, $C_{xx,xy}$ etc.[682, 683, 684]) is the preferred format for reporting.

```
STATIC POLARIZABILITY TENSOR (Quadrupole/Quadrupole)
The order in each direction is XX, YY, ZZ, XY, XZ, YZ
Method
                : SCF
Method : SCr
Type of density : Electron Density
Type of derivative : Quadrupolar Field (Direction=X)
Multiplicity : 1
Irrep
Relativity type
Basis
                 : AO
The raw cartesian tensor (atomic units):
                                        8.017351959 -0.002591466
     60.656194448 8.024072323
    0.000000801
                     -0.000000184
                                        12.837825709
                   55.906127614
-0.00000529
     8.024072323
                                                           -6.821368242
→ -0.000000954
                     12.837825709
                                        55.938851507
     8.017351959
                                                           6.815300773
   0.000000232
                     0.000000422
    -0.002591466
                      -6.821368242
                                         6.815300773
                                                          16.716647772
→ 0.00000169
                     -0.00000030
                      -0.000000954
     0.000000801
                                         0.000000232
                                                           0.000000169
→ 16.715850196
                     6.818791255
                   -0.00002
21.534628724
     -0.00000184
                      -0.000000529
                                         0.000000422
                                                           -0.000000030
    6.818791255
diagonalized tensor:
    11.893291534 13.500711
76.753292120
                      13.566719080
                                         26.357187387
                                                           46.234564137
   52.663246003
Orientation:
     -0.00000018
                      -0.000019986
                                         -0.00000013
                                                           -0.001433194
```

\hookrightarrow				(continued from previou	1.0.7
	0.817436692	0.576016666			
	0.00000006	0.219691224	0.00000038	0.673107563	_
\hookrightarrow	-0.405967809	0.577799371			
	0.00000008	-0.219450737	-0.000000021	-0.671194117	_
\hookrightarrow	-0.408640606	0.578232381			
	-0.00000035	0.950566746	-0.00000034	-0.310519906	_
\hookrightarrow	-0.000497156	-0.000034103			
	0.816443231	0.00000038	0.577425709	-0.00000037	_
\hookrightarrow	0.00000027	0.00000000			
	-0.577425709	-0.00000003	0.816443231	-0.00000036	_
\hookrightarrow	0.00000002	-0.00000003			
Isot	ropic polarizak	ility : 37.91138			
 STAT	TO TRACFIESS DO	 DLARIZABILITY TENSOR (Q			
SIAI	IC IRACELESS PC	LARIZABILIII IENSOR (Q	uadrupoie/Quadrupoie)		
The	order in each o	lirection is XX, YY, ZZ	XY X7. Y7.		
1110	01001 111 00011 0	11, 22,	,,,		
Meth	od	: SCF			
Type	of donaity				
	: Or density	: Electron Density			
Type	_	: Electron Density : Quadrupolar Field (D:	irection=X)		
	_	_	irection=X)		
	of derivative iplicity	: Quadrupolar Field (D	irection=X)		
Mult Irre	of derivative iplicity	: Quadrupolar Field (D: 1	irection=X)		
Mult Irre	of derivative iplicity op	: Quadrupolar Field (D. : 1 : 0	irection=X)		
Mult Irre Rela Basi	e of derivative iplicity ep tivity type .s	<pre>: Quadrupolar Field (D. : 1 : 0 : : AO</pre>	irection=X)		
Mult Irre Rela Basi	e of derivative iplicity ep tivity type s	<pre>: Quadrupolar Field (D. : 1 : 0 : : AO ensor (atomic units):</pre>			
Mult Irre Rela Basi	e of derivative iplicity ep tivity type .s	<pre>: Quadrupolar Field (D. : 1 : 0 : : AO</pre>	irection=X) -13.171591878	0.000221134	ı
Mult Irre Rela Basi	e of derivative iplicity ep tivity type s raw cartesian t 26.331642600 0.000000581	<pre>: Quadrupolar Field (D. : 1 : 0 : : AO ensor (atomic units):</pre>	-13.171591878		1
Mult Irre Rela Basi	e of derivative iplicity Ep tivity type s raw cartesian t 26.331642600 0.00000581 -13.160050722	<pre>: Quadrupolar Field (D. : 1 : 0 : : AO ensor (atomic units):</pre>		0.000221134 -5.113861448]]
Mult Irre Rela Basi	e of derivative iplicity Ep tivity type s raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735	<pre>: Quadrupolar Field (D. : 1 : 0 :</pre>	-13.171591878 -9.573838294	-5.113861448	1
Mult Irre Rela Basi The	e of derivative iplicity Ep tivity type .s raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878	: Quadrupolar Field (D.: 1 : 0 : 0 : : AO ensor (atomic units): -13.160050722 -0.000000065 22.733889017 -0.000000324 -9.573838294	-13.171591878		
Mult Irre Rela Basi The	e of derivative iplicity Ep tivity type .s raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878 0.000000154	: Quadrupolar Field (D.: 1 : 0 : 0 : : AO ensor (atomic units): -13.160050722 -0.000000065 22.733889017 -0.000000324 -9.573838294 0.000000389	-13.171591878 -9.573838294 22.745430172	-5.113861448 5.113640314	
Mult Irre Rela Basi The	raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878 0.000000154 0.000221134	: Quadrupolar Field (D.: 1 : 0 : 0 : : AO ensor (atomic units): -13.160050722 -0.000000065 22.733889017 -0.000000324 -9.573838294 0.000000389 -5.113861448	-13.171591878 -9.573838294	-5.113861448	
Mult Irre Rela Basi The	raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878 0.000000154 0.000221134 0.000000127	: Quadrupolar Field (D.: 1 : 0 : 0 : : AO ensor (atomic units): -13.160050722 -0.000000065 22.733889017 -0.000000324 -9.573838294 0.000000389 -5.113861448 -0.000000022	-13.171591878 -9.573838294 22.745430172 5.113640314	-5.113861448 5.113640314 12.537485829	u
Mult Irre Rela Basi The	raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878 0.000000154 0.000221134 0.000000127 0.000000581	: Quadrupolar Field (D.: 1 : 0 : 0 : : AO ensor (atomic units): -13.160050722 -0.000000065 22.733889017 -0.000000324 -9.573838294 0.000000389 -5.113861448 -0.000000022 -0.000000735	-13.171591878 -9.573838294 22.745430172	-5.113861448 5.113640314	u
Mult Irre Rela Basi The	raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878 0.000000154 0.000221134 0.000000127 0.000000581 12.536887647	: Quadrupolar Field (D. : 1 : 0 : 1 : AO ensor (atomic units):	-13.171591878 -9.573838294 22.745430172 5.113640314 0.000000154	-5.113861448 5.113640314 12.537485829 0.000000127	
Mult Irre Rela Basi The	raw cartesian t 26.331642600 0.000000581 -13.160050722 -0.000000735 -13.171591878 0.000000154 0.000221134 0.000000127 0.000000581	: Quadrupolar Field (D.: 1 : 0 : 0 : : AO ensor (atomic units): -13.160050722 -0.000000065 22.733889017 -0.000000324 -9.573838294 0.000000389 -5.113861448 -0.000000022 -0.000000735	-13.171591878 -9.573838294 22.745430172 5.113640314	-5.113861448 5.113640314 12.537485829	

1 Note

• Like the quadrupole moments themselves, the dipole-quadrupole and quadrupole-quadrupole polarizabilities depend on the gauge origin of the <code>%elprop</code> module. The latter can be changed using the <code>Origin</code> keyword in <code>%elprop</code>; see section <code>Electrical Properties</code> - <code>Electric Moments and Polarizabilities</code>.

At the SCF level, dynamic (frequency-dependent) dipole polarizabilities can be computed using either purely real or purely imaginary frequencies.

```
%elprop
polar 1
freq_r 0.08 # purely real frequencies
#freq_i 0.08 # purely imaginary frequencies
end
```

In the example above, the dynamic dipole polarizability tensor for a single real frequency of 0.8 a.u. is computed. For every frequency, linear response equations must be solved for all component of the perturbation operator (3 Cartesian components of the electric dipole). Note that imaginary-frequency polarizabilities are computed with the same costs as real-frequency polarizabilities. By a simple contour integration they can be used to compute dispersion coefficients.

For methods that do not support analytic polarizabilities, one can calculate numeric dipole-dipole and quadrupole-quadrupole polarizabilities, either by finite differentiation of dipole/quadrupole moments with respect to a finite dipole/quadrupole electric field, or by double finite differentiation of the total energy with respect to a finite dipole/quadrupole electric field. The latter can be done using compound scripts (see *Compound*, *Compound Examples*).

At the MP2 level, polarizabilities can currently be calculated analytically using the RI (*RI-MP2 and Double-Hybrid DFT Response Properties*) or DLPNO (*Local MP2 Second Derivatives and Response Properties*) approximations or in all-electron canonical calculations, but for canonical MP2 with frozen core orbitals the dipole moment has to be differentiated numerically in order to obtain the polarizability tensor. In general in such cases, you should keep in mind that tight SCF convergence is necessary in order to not get too much numerical noise in the second derivative. Also, you should experiment with the finite field increment in the numerical differentiation process.

As an example, the following Compound job can be used to calculate the seminumeric polarizability at the MP2 level (replacing the now deprecated usage of Polar 2):

```
*xyz 0 1
0 0.0000000000000 0.05591162058341 0.05591162058342
H 0.0000000000000 -0.06629333722358 1.01038171664016
H 0.000000000000 1.01038171664017 -0.06629333722358

*

*Compound "semiNumericalPolarizability.cmp"
with
method = "MP2";
basis = "aug-cc-pVDZ cc-pVDZ/C";
restOfInput = "VeryTightSCF PModel NoFrozenCore";
end
```

with the file semiNumericalPolarizability.cmp containing:

```
# Authors: Dimitrios G. Liakos and Frank Neese
# Date : March-May of 2024
# This is a compound script that calculates the
  raw porarizability tensor semi-numerically
  using the dipole moments.
#
 The idea is the following:
 1. Calculate dipole moment in the field free case
#
# 2. Loop over directions I=X,Y,Z
    - put a small E-field in direction I+Delta
     - Solve equations to get the dipole moment D+
    - put s small E-field in direction I-Delta
    - Solve equations to get the dipole moment D-
    - Polarizability alpha (I, J). (D+(I)-D-(I))/(2Delta)
    - Diagonalize to get eigenValues, eigenVectors, a_iso
# 3. Print polarisability
# NOTE: We use the last dipole_moment found in the file. If a specific
       one is needed the 'myProperty' option should be accordingly
#
#
       adjusted and remove the 'property_Base = true' option.
```

```
# NOTE: This is not the most general version. It is adjusted for testsuite
# with 'method' and 'mp2' blocks.
      ----- Variables
# --- Variables to be adjusted (e.g. using 'with' -
Variable molecule = "h2o.xyz";  # xyz file with coordinates
Variable charge
                     = 0;
Variable mult
Variable method
Variable basis
                     = "HF";
                     = " " ;
Variable basis
Variable restOfInput = "NoFrozenCore VeryTightSCF";
Variable E_Field = 0.0001; # Size of Electric field
Variable myProperty = "Dipole_Moment_Total";
Variable removeFiles = true;
                                        # Remove files in the end of the.
⇔calculation
# ----- Rest of the variables -----
Variable D_Free, D_Minus, D_Plus, a[3][3]; #dipole moment and polarizability
Variable aEigenValues[3], aEigenVectors[3][3], a_iso;
Variable FFieldStringPlus, FFieldStringMinus;
Variable res = -1;
# Field Free
New Step
 !&{method} &{basis} &{restOfInput}
  %Method
   z_tol 1e-8
  %MP2
  Density Relaxed
 #*xyzFile &{charge} &{mult} &{molecule}
Step_End
res = D_Free.readProperty(propertyName=myProperty, property_Base=true);
# Loop over the x, y, z directions and create the appropriate string
for direction from 0 to 2 Do
 #Create the appropriate direction oriented field string
 if (direction = 0) then #( X direction)
   write2String(FFieldStringPlus, "%lf, 0.0, 0.0", E_Field);
write2String(FFieldStringMinus, "-%lf, 0.0, 0.0", E_Field);
  else if (direction = 1) then \# ( Y direction)
   write2String(FFieldStringPlus, " 0.0, %lf, 0.0", E_Field);
write2String(FFieldStringMinus, " 0.0, -%lf, 0.0", E_Field);
                                   #( Z direction)
   write2String(FFieldStringPlus, " 0.0, 0.0, %lf", E_Field);
    write2String(FFieldStringMinus, " 0.0, 0.0, -%lf", E_Field);
  EndIf
  # Perform the calculations.
  # First the plus (+) one
  # -----
  ReadMOs(1);
  New Step
   !&{method} &{basis} &{restOfInput}
```

```
%SCF
    EField = &{FFieldStringPlus}
   End
   %Method
    z_tol 1e-8
   End
   %MP2
    Density Relaxed
   End
 Step_End
 res = D_Plus.readProperty(propertyName=myProperty, property_Base=true);
 # And the minus (-) one
 ReadMOs(1);
 New Step
   !&{method} &{basis} &{restOfInput}
    EField = &{FFieldStringMinus}
   %Method
    z_tol 1e-8
   End
   %MP2
    Density Relaxed
   End
 Step_End
 res = D_Minus.readProperty(propertyName=myProperty, property_Base=true);
 # Construct and store SCF polarizability
 # -----
 a[direction][0] = (D_Plus[0]-D_Minus[0])/(2*E_Field);
 a[direction][1] = (D_Plus[1]-D_Minus[1])/(2*E_Field);
 a[direction][2] = (D_Plus[2]-D_Minus[2])/(2*E_Field);
EndFor
# Diagonalize
a.Diagonalize(aEigenValues, aEigenVectors);
# Do some printing
print( "\n\n");
print( " ----\n");
print( "
                COMPOUND
print( " Semi analytical calculation of polarizability\n");
print( "
print( " Method: %s\n", method);
print( " Basis : %s\n", basis);
print( " The electric field perturbation used was: %.5lf a.u.\n", E_Field);
print( " \n\n");
print( " ----\n");
print( " Raw electric semi-analytical polarizability tensor\n");
print( " -----
For i from 0 to 2 Do
 print("%13.81f %13.81f %13.81f\n", a[i][0], a[i][1], a[i][2]);
```

```
EndFor
print( " -----\n");
print("\n");
print( " -----
print( " Raw electric semi-analytical polarizability Eigenvalues\n");
print("%13.81f %13.81f %13.81f\n", aEigenValues[0], aEigenValues[1],_
→aEigenValues[2]);
print( " -----
print("\n");
print( " Raw electric semi-analytical polarizability Eigenvectors\n");
print( " -----\n");
For i from 0 to 2 Do
 print("%13.81f %13.81f %13.81f\n", aEigenVectors[i][0], aEigenVectors[i][1],_
→aEigenVectors[i][2]);
EndFor
print( "\n a isotropic value : %.5lf\n",_
# Maybe remove unneccesary files
if (removeFiles) then
 sys_cmd("rm *_Compound_* *.bas* ");
EndIf
End
```

For more details on Compound jobs in general, see Compound.

For other correlation methods, where not even relaxed densities are available, only a fully numeric approach (using compounds scripts) is possible and requires obnoxiously tight convergence.

Note that polarizability calculations have higher demands on basis sets. A rather nice basis set for this property is the Sadlej one (see *Orbital Basis Sets*). In relation to electric properties, it might be interesting to know that it is possible to carry out finite electric field calculations in ORCA. See section *Finite Electric Fields* for more information on such calculations.

The static hyperpolarizability tensor is calculated as an analytical third derivative of the energy with respect to three dipole perturbations. It is currently available for SCF methods (HF, DFT). For example, consider the input:

The hyperpolarizability can be found printed in the output as

```
STATIC HYPERPOLARIZABILITY TENSOR
Type of density : Electron Density
Type of derivative : Electric Field (Direction=X)
Multiplicity : 1
Irrep
Basis
                          : AO
  The raw Cartesian tensor (atomic units):
       ( x x x ):
( x x y ):
( x x z ):
                                 -0.000000
                                 -0.000000
                                  7.076938
       ( x y x ):
( x y y ):
                                  -0.000000
                                  -0.000000
                                 -0.000000
       ( x y z ):
                                  7.076938
      ( x z x ):

( x z y ):

( x z z ):

( y x x ):

( y x y ):

( y x z ):

( y y x ):

( y y z ):

( y z x ):

( y z y ):

( z x x ):

( z x z ):

( z y x ):

( z y z ):

( z y z ):

( z z x y ):

( z z y z ):
       ( x z x ):
                                -0.000000
                                  -0.000000
                                  -0.000000
                                  -0.000000
                                 -0.000000
                                 -0.00000
                                  0.000000
                                49.371503
                                 -0.000000
                                49.371503
                                  0.000000
                                  7.076938
                                -0.000000
                                -0.000000
                                 -0.000000
                                49.371503
                                  0.000000
                                  -0.000000
                                   0.000000
       ( z z y ):
                                  44.368713
       ( z z z ):
```

5.20.1 Atomic Dipole Moment and Polarizabilities

ORCA includes a mechanism by which electric properties can be decomposed into atomic contributions. In a nutshell, the method relies on the partitioning of real space into atomic regions. These regions are defined by a smooth weighting function for atom A

$$f_A\left(\mathbf{r}
ight) = rac{arrho_A^{(0)}(\mathbf{r})}{\sum_B arrho_B^{(0)}(\mathbf{r})}$$

The quantities $\varrho_A^{(0)}(r)$ are spherically symmetric atomic densities (expanded in a reasonably large set of Gaussian s-functions) for the neutral atoms that exit pretabulated in the code.

Using these partitioning functions, an atomic multipole moment is defined as:

$$Q_l^m(A) = \sum_{\mu\nu} P_{\mu\nu} \int \mu(\mathbf{r}) \widehat{q}_l^m \nu(\mathbf{r}) f_A(\mathbf{r}) d\mathbf{r}$$

Here **P** is the density matrix and \hat{q}_l^m the operator for a multipole moment of order l with component m. For example, for the dipole moment l=1 and m=0,+1,-1. Likewise, for a quadrupole moment l=2, m=2,1,0,-1,-2 and so on. The

partitioned integration over space is carried out by numerical integration. In the limit of exact numerical integration, the construction guarantees that the atomic moments sum to the overall multipole moment. Because numerical integration introduces a small error, the accumulated moments

$$Q_l^m = \sum_A Q_l^m(A)$$

Will show small deviation from the analytically integrated moments that the program also calculates prints.

Clearly, the same strategy can be used for response properties. For example, for polarizabilities, one decomposes it into the atomic components as follows:

$$\alpha_{KL}(A) = -\sum_{\mu\nu} \frac{\partial P_{\mu\nu}}{\partial F_K} \int \mu(\mathbf{r}) \mathbf{r}_L \nu(\mathbf{r}) f_A(\mathbf{r}) d\mathbf{r}$$

Here F is an external electric field and the derivative of the density matrix is the response density with respect to this

In order to trigger this calculation use:

```
%elprop
Polar true
PolarAtom true
Dipole true
DipoleAtom true
Quadrupole true
QuadrupoleAtom true
end
```

1 Note

- · Atomic electric properties are available presently at the SCF level for dipole moments, quadrupole moments and polarizabilities
- · The dipole moment is well-known to be gauge dependent for charged molecules. Hence, the way the dipole moment is decomposed here, the atomic dipole moments refer to the global origin and there is a gauge dependence for charged atoms. Consequently, one needs to proceed with some caution in the interpretation of these local dipole moments.

PAPER:

• Grigorash, D.; Müller, S.; Heid, E.; Neese, F.; Liakos, D.; Riplinger, C.; Garcia-Rates, M.; Paricaud, P. Erling H.S.; Smirnova, I.; Yan, W. A comprehensive approach to incorporating intermolecular dispersion into the openCOSMO-RS model. Part 2: Atomic polarizabilities, 2025, submitted

5.20.2 Elprop Keywords

Calculation of first order (electric dipole and quadrupole moments) and second order (polarizabilities) electric properties can be requested in the %elprop input block. The second order properties can be calculated through the solution of the CP-SCF (see CP-SCF Options) or CP-CASSCF (see CASSCF Linear Response) equations. Details are shown below:

```
%elprop
 Dipole
         true
 Quadrupole true
 PolarVelocity true # polarizability w.r.t. velocity perturbations
 PolarDipQuad true # dipole-quadrupole polarizability
 PolarQuadQuad true # quadrupole-quadrupole polarizability
```

```
Hyperpol
             true # the dip/dip/dip hyperpolarizability
freq_r 0.00
              # purely real frequency (default: static calculation)
              # purely imaginary frequency (default: static calculation)
freq_i 0.00
              # CG(conjugate gradient)
Solver CG
              # other options: DIIS or POPLE(default)
MaxDIIS 5
             # max. dimension of DIIS method
Shift 0.2
             # level shift used in DIIS solver
       1e-3
             # Convergence of the CP-SCF equations
              # (norm of the residual)
MaxIter 64
              # max. number of iterations in CPSCF
PrintLevel 2
Origin CenterOfElCharge # center of electronic charge
        CenterOfNucCharge # center of nuclear charge
        CenterOfSpinDens # center of spin density
        CenterOfMass # center of mass (default)
                         # position of atom N (starting at 0)
                         # explicit position of the origin
       X, Y, Z
                         # in coordinate input units (Angstrom by default)
end
```

The most efficient and accurate way to calculate the polarizability analytically is to use the coupled-perturbed SCF method. The most time consuming and least accurate way is the numerical second derivative of the total energy. Note that the numerical differentiation requires: (a) tightly or even very tightly converged SCF calculations and (b) carefully chosen field increments. If the field increment is too large then the truncation error will be large and the values will be unreliable. On the other hand, if the field increment is too small the numerical error associated with the finite difference differentiation will get unacceptably large up to the point where the whole calculation becomes useless.

5.21 Nuclear Magnetic Resonance (NMR) Parameters

5.21.1 NMR Chemical Shifts

NMR chemical shifts at the HF, DFT (standard GGA and hybrid functionals), CASSCF, as well as the RI- and DLPNO-MP2 and double-hybrid DFT levels (see section MP2 Level Magnetic Properties and references therein), plus Coupled Cluster (GIAO-CCSD, GIAO-CCSD(T), see AUTOCI Response Properties via Analytic Derivatives) can be obtained from the EPR/NMR module of ORCA. For the calculation of the NMR shielding tensor the program utilizes Gauge Including Atomic Orbitals (GIAOs - sometimes also referred to as London orbitals). [685, 686, 687] In this approach, field dependent basis functions are introduced, which minimizes the gauge origin dependence and ensures rapid convergence of the results with the one electron basis set. [688] Note that GIAOs are NOT currently available with CASSCF linear response and a gauge origin must be provided in the %eprnmr block (see CASSCF Linear Response). GIAOs for CASSCF response are coming soon to ORCA! A complete list of possible keywords for the eprnmr module can be found in EPRNMR - keywords for magnetic properties.

The use of the chemical shift module is simple:

```
# Ethanol - Calculation of the NMR chemical shieldings at the HF/SVP level of.
 →theory
! RHF SVP Bohrs NMR
* xyz 0 1
C
        -1.22692181
                       0.24709455
                                   -0.00000000
С
        -0.01354839
                      -0.54677253
                                     0.00000000
Н
        -2.09280406
                      -0.41333631
                                     0.00000000
Η
        -1.24962478
                       0.87541936
                                   -0.88916500
Н
        -1.24962478
                       0.87541936
                                     0.88916500
0
         1.09961824
                       0.30226226
                                     -0.00000000
Н
         0.00915178
                       -1.17509696
                                      0.88916500
```

```
H 0.00915178 -1.17509696 -0.88916500

H 1.89207683 -0.21621566 0.000000000

*
```

The output for the shieldings contains detailed information about the para- and diamagnetic contribution, the orientation of the tensor, the eigenvalues, its isotropic part etc. For each atom, an output block with this information is given:

```
Nucleus OC:
Diamagnetic contribution to the shielding tensor (ppm) :
          209.647 -10.519 0.000
                      215.858
                                      0.000
          -26.601
          -0.000
                        0.000
                                    200.382
Paramagnetic contribution to the shielding tensor (ppm):
          59.273 18.302 -0.000
          13.380
                         6.063
                                     -0.000
           0.000
                        -0.000
                                     -2.770
Total shielding tensor (ppm):
          268.920 7.783
                                     -0.000
                                     -0.000
          -13.220
                       221.921
                                    197.611
           0.000
                         0.000
Diagonalized sT*s matrix:
              200.382
                              214.507
                                              210.998 iso=
                                                              208.629
sDSO
                                                               20.855
sPSO
               -2.770
                                7.279
                                              58.057 iso=
Total
              197.611
                              221.786
                                              269.055 iso=
                                                              229.484
```

Note that all units are given in ppm and the chemical shieldings given are *absolute* shieldings (see below). At the end of the atom blocks, a summary is given with the isotropic shieldings and the anisotropy [689] for each nucleus:

Nucleus	Element	Isotropic	Anisotropy
0	С	229.484	59.356
1	С	227.642	62.878
2	Н	56.015	12.469
3	Н	55.460	15.284
4	Н	55.460	15.284
5	0	334.125	110.616
6	Н	47.337	27.101
7	Н	47.337	27.101
8	Н	64.252	32.114

Thus, the absolute, isotropic shielding for the ¹³C nuclei are predicted to be 229.5 and 227.6 ppm and for ¹⁷O it is 334.1 ppm. While basis set convergence using GIAOs is rapid and smooth, it is still recommended to do NMR calculations with basis sets including tight exponents, such as the purpose-built pcSseg-n. However, TZVPP or QZVP should be sufficient in most cases. [690, 691]

An important thing to note is that in order to compare to experiment, a standard molecule for the type of nucleus of interest has to be chosen. In experiment, NMR chemical shifts are usually determined relative to a standard, for example either CH₄ or TMS for proton shifts. Hence, the shieldings for the molecule of interest and a given standard molecule are calculated, and the relative shieldings are obtained by subtraction of the reference value from the computed value. It is of course important that the reference and target calculations have been done with the same basis set and functional. This also helps to benefit from error cancellation if the standard is chosen appropriately (one

option is even to consider an "internal standard" - that is to use for example the 13 C shielding of a methyl group inside the compound of interest as reference).

Let us consider an example - propionic acid (CH₃-CH₂COOH). In databases like the AIST (http://sdbs.db.aist.go.jp) the 13 C spectrum in CDCl₃ can be found. The chemical shifts are given as δ_1 = 8.9 ppm, δ_2 = 27.6 ppm, δ_3 = 181.5 ppm. While intuition already tells us that the carbon of the carboxylic acid group should be shielded the least and hence shifted to lower fields (larger δ values), let's look at what calculations at the HF, BP86 and B3LYP level of theory using the SVP and the TZVPP basis sets yield:

method	σ_1	σ_2	σ_3
HF/SVP	191.7	176.6	23.7
HF/TZVPP	183.5	167.1	9.7
B86/SVP	181.9	165.8	26.5
B86/TZVPP	174.7	155.5	7.6
B3LYP/SVP	181.8	165.8	22.9
B3LYP/TZVPP	173.9	155.0	2.9

Looking at these results, we can observe several things - first of all, the dramatic effect of using too small basis sets, which yields differences of more than 10 ppm. Second, the results obviously change a lot upon inclusion of electron correlation by DFT and are functional dependent. Last but not least, these values have nothing in common with the experimental ones (they change in the wrong order), as the calculation yields *absolute shieldings* like in the table above, but the experimental ones are *relative shifts*, in this case relative to TMS.

In order to obtain the relative shifts, we calculate the shieldings σ_{TMS} of the standard molecule (TMS HF/TZVPP: 194.1 ppm, BP86/TZVPP: 184.8 ppm, B3LYP/TZVPP: 184.3 ppm) and by using $\delta_{mol} = \sigma_{ref} - \sigma_{mol}$ we can evaluate the chemical shifts (in ppm) and directly compare to experiment:

method	δ_1	δ_2	δ_3
HF/TZVPP	10.6	27.0	184.4
B86/TZVPP	10.1	29.3	177.2
B3LYP/TZVPP	10.4	29.3	181.4
Exp.	8.9	27.6	181.5

A few notes on the GIAO implementation in ORCA are in order. The use of GIAOs lead to some fairly complex molecular one- and two-electron integrals and a number of extra terms on the right hand side of the coupled-perturbed SCF equations. The two-electron contributions in particular can be time consuming to calculate. Thus, the RIJK, RIJDX, and RIJCOSX approximations were implemented and tested.[692] By default, the approximation used for the SCF is also applied to the GIAO integrals, but the latter can be changed using the GIAO_2el keyword in the eprnmr input block (see section *EPRNMR - keywords for magnetic properties*). Note that, while the default COSX grids are typically sufficiently accurate for chemical shifts, the use of defgrid3 is recommended for special cases or post-HF calculations.

The user can finely control for which nuclei the shifts are calculated (although this will not reduce the computational cost very much, which is dominated by the CP-SCF equations for the magnetic field). This works in exactly the same way as for the hyperfine and quadrupole couplings described in the next section. For example:

```
! B3LYP def2-TZVP TightSCF

* int 0 1
C 0 0 0 0 0 0
C 1 0 0 1.35 0 0
H 1 2 0 1.1 120 0
H 1 2 3 1.1 120 180
H 2 1 3 1.1 120 0
H 2 1 3 1.1 120 180
*
```

```
%eprnmr
Ori = GIAO
Nuclei = all C { shift }
Nuclei = all H { shift }
end
```

NMR chemical shifts are also implemented in combination with implicit solvent models, hence the NMR keyword can be combined with the CPCM input block. Note that for calculations including implicit solvent, it is highly recommended to also optimize the geometries using the same model. Regardless, explicit solvent—solute interactions observable in NMR (e.g. H-bonds), cannot be modelled with such a model: solvent molecules must be included explicitly in the calculation.

5.21.2 NMR Spin-Spin Coupling Constants

The indirect spin-spin coupling constants observed in NMR spectra of molecules in solution consist of four contributions: The diamagnetic spin orbit term:

$$\hat{H}_{DSO} = \frac{1}{2} \sum_{ikl} \frac{(\vec{M}_k \times \vec{r}_{ik})(\vec{M}_l \times \vec{r}_{il})}{r_{ik}^3 r_{il}^3}$$
(5.91)

The paramagnetic spin orbit term:

$$\hat{H}_{PSO} = \sum_{ik} \frac{\vec{M}_k \, \vec{l}_{ik}}{r_{ik}^3} \tag{5.92}$$

The Fermi contact term:

$$\hat{H}_{FC} = \frac{8\pi}{3} \sum_{ik} \delta(r_i - r_k) \,\mathbf{m}_k \tag{5.93}$$

And the spin dipole term:

$$\hat{H}_{SD} = \sum_{ik} \mathbf{m}_k^T \frac{3 \mathbf{r}_{ik} \mathbf{r}_{ik}^T - r_{ik}^2}{r_{ik}^5} \mathbf{s}_i$$
 (5.94)

While the Fermi contact term is usually the most significant, all contributions can be computed at the HF and DFT level of theory using ORCA. For this purpose, the keyword ssall has to be invoked in the eprnmr input block, while each of the four terms can be requested using ssdso, sspso, ssfc, and sssd, respectively. For example:

Results will be given in Hz. Note that the default isotopes used might not be the ones desired for the calculation of NMR properties, so it is recommended to define the corresponding isotopes using the ist flag. It is possible to also print the reduced coupling constants K (in units of $10^{19} \cdot T \cdot J^{-2}$), which are independent of the nuclear isotopes, using the flag PrintReducedCoupling=True.

The CP-SCF equations must be solved for one of the nuclei in each pair and are the bottleneck of the computation. Therefore, spin-spin coupling constants are calculated only between nuclei within a certain distance of eachother (5 Ångstrom by default). The latter can be changed using the SpinSpinRThresh keyword.

If mulitple nuclides are requested, it is also possible to select only certain element pairs (e.g. only C—H and H—H, without C—C) using the SpinSpinElemPairs keyword. Analogously, the SpinSpinAtomPairs keyword selects the actual pairs of nuclei to consider. The union of the latter two options is used to *reduce* the selection made using the Nuclei input, after which SpinSpinRThresh is applied.

Here is another example illustrating these additional options:

```
! B3LYP EPR-II
 xyz 0 1
                0.247095 -0.000000
-0.546773 0.000000
-0.413336 0.000000
С
      -1.226922
С
      -0.013548
Н
      -2.092804
                0.875419
                             -0.889165
Η
      -1.249625
                  0.875419 0.889165
      -1.249625
Н
       1.099618
                 0.302262 -0.000000
0
       0.009152 -1.175097 0.889165
Н
Н
       0.009152 -1.175097 -0.889165
Н
       1.892077 -0.216216 0.000000
%eprnmr
 nuclei = all C { ssall, ist = 13 };
 nuclei = all H { ssall, ist = 1 };
 nuclei = all 0 { ssall, ist = 17 };
 PrintReducedCoupling true
 SpinSpinRThresh 3.0 # Angstrom
 SpinSpinElemPairs {C C} {O *} # "*" means any element
 SpinSpinAtomPairs {8 *}
                              # indices start from 0
  # Final selection:
  # C 0 - C 1
  # C 0 - O 5
  # C 1 - O 5
  # C 1 - H 8
  # H 3 - O 5
  # H 4 - O 5
  # 0 5 - H 6
  # 0 5 - H 7
  # O 5 - H 8
  # H 6 - H 8
  # H 7 - H 8
end
```

5.21.3 Simulating NMR Spectra

From the computed NMR shieldings and spin-spin coupling constants, the coupled NMR spectra can be simulated. The most typical NMR experiments are decoupled ^{13}C and coupled ^{1}H spectra, so we will focus on these here. For simulating a full NMR spectrum, we will use ethyl crotonate as an example, and three steps are required: 1) computation of the spin-spin coupling constants, 2) computation of the chemcial shieldings and 3) simulation of the spectrum using a spin-Hamiltonian formalism. Note that these steps can be carried out independently and different levels of theory can be used for shieldings and couplings and the order of steps 1 and 2 doesn't matter.

Furthermore, if the spectra are to be simulated with TMS as reference, the shieldings for TMS are required (the hydrogen and carbon values in this case are 31.77 and 188.10 ppm, respectively). Here is the input for the calculation for the coupling constants, which is named ethylcrotonate_sscc.inp:

```
! PBE pcJ-3 autoaux tightscf
*xyzfile 0 1 ethylcrotonate.xyz
(continues on next page)
```

```
%eprnmr
Nuclei = all H {ssall}
end
```

The spin-spin coupling constants will be stored in the file ethylcrotonate_sscc.property.txt, which the simulation of the NMR spectrum will need to read. The NMR shieldings and will be computed in the next step, for which the input ethylcrotonate_nmr.inp looks like this:

```
!TPSS pcSseg-3 autoaux tightscf NMR

*xyzfile 0 1 ethylcrotonate.xyz
```

The final NMR spectrum simulation is carried out using orca_nmrspectrum, which requires a separate input file ethylcrotonate.nmrspec which looks like this (note the required final END statement):

```
NMRShieldingFile = "ethylcrotonate_nmr"
                                           #property file for shieldings
NMRCouplingFile = "ethylcrotonate_sscc"
                                           #property file for couplings
NMRSpecFreq = 80.00
                                           #spectrometer freq [MHz] (default 400)
PrintLevel = 0
                                           #PrintLevel for debugging info
NMRCoal = 1.0
                                           #threshold for merged lines [Hz]_
\hookrightarrow (default 1)
NMRREF[1] 31.77
                                           #shielding for 1H reference [ppm]
NMRREF[6] 188.10
                                           #shielding for 13C reference [ppm]
NMREquiv
                                           #lists of NMR-equivalent nuclei
1 {13 14 15} end
                                           #H 13,14,15 are equivalent (methyl)
                                           #H 16 and 17 equivalent (ethyl)
2 {16 17} end
3 {8 10 11} end
                                           #H 8,10,11 again equivalent methyl
end
                                           #end equiv nucl block
END
                                           #essential end of input
```

and contains the following keywords:

NMRShieldingFile and NMRCouplingFile denote the .property.txt files from which the shielding tensors and coupling constants will be read by the NMR spectrum module. If this line is not given, the program will exepect the shieldings or couplings in the property file of the current calculation.

NMRSpecFreq The NMR spectrometer frequency in MHz is decisive for the looks of the spectrum as shieldings are given in ppm and couplings are given in Hz. Default is 400 MHz.

NMRCoal If two lines are closer than this threshold (given in Hz) then the module will coalesce the lines to one line with double intensity. Default it 1 Hz.

NMRREF [X] Reference value for the absolute shielding of element X used in the relative shifts of the simulated spectrum. Typically, these are the absolute shielding values from a separate calculation of TMS, for example, and will be zero ppm in the simulated spectrum.

NMREquiv The user has to specify groups of NMR equivalent nuclei. These are typically atoms which interchange on the NMR timescale, like methyl group protons. The shieldings and couplings will be averaged for each group specified by the user.

with this input, orca_nmrspectrum is called with two arguments, the first one is a gbw file which contains all informations about the molecule, typically this is the gbw file of the nmr or the sscc calculation, and the name of the input file given above:

```
orca_nmrspectrum ethylcrotonate_nmr.gbw ethylcrotonate.nmrspec > output
```

If this calculation is carried out, the NMR spectrum module will read the files

ethylcrotonate_sscc.property.txt and ethylcrotonate_nmr.property.txt, extract the shieldings and couplings, average the equivalent nuclei and set up an effective NMR spin Hamiltonian for each nucleus:

$$H_{NMR}(p,q) = \sigma_p \delta_{pq} + J_{pq} I_p I_q. \tag{5.95}$$

***** Caution

This includes all nuclei this nuclear spin couples to and should not contain too many spins (see SpinSpinRThres), as the spin Hamiltonian for each atom and the nuclei it couples to is diagonalized brute force. Afterwards, all spin excitations are accumulated and merged into the final spectrum for each element. For ethyl crotonate the NMR spectrum output looks like this:

```
NMR Peaks for atom type 1, ref value 31.7700 ppm :
Atom shift[ppm] rel.intensity
               2.34 9.00
                     2.36 9.00
     8 2.36 9.00

8 2.25 9.00

8 2.27 9.00

9 6.34 1.00

9 6.36 3.00

9 6.38 3.00

9 6.41 1.00

9 6.14 1.00

9 6.16 3.00

9 6.19 3.00

9 6.21 1.00

12 7.95 1.00

12 7.85 3.00

12 7.75 4.00
    12
    12
                     7.75
                                      4.00

    12
    7.75
    4.00

    12
    7.65
    4.00

    12
    7.56
    3.00

    12
    7.47
    1.00

    13
    1.71
    9.00

    13
    1.61
    18.00

    13
    4.56
    4.00

    16
    4.46
    12.00

    16
    4.37
    12.00

    16
    4.27
    4.00

    12
   16
                      4.27 4.00
 NMR Peaks for atom type 6, ref value 188.1000 ppm :
Atom shift[ppm] rel.intensity
     2 25.70 1.00
3 155.15 1.00
4 19.96 1.00
5 68.91 1.00
6 174.39 1.00
7 130.29 1.00
                  130.29
                                         1.00
 NMR Peaks for atom type 8, ref value 104.8826 ppm :
Atom shift[ppm] rel.intensity
     0 0.00 5.00
1 149.74 5.00
```

The first column denotes the atom number of the nucleus the signal arises from, the second column denotes the line position in ppm and the third line denotes the relative intensity of the signal. For oxygen, no reference value was given, so the program will autmatically set the lowest value obtained in the calculation as reference value.

Using gnuplot, for example, to plot the spectrum, the following plots for ${}^{13}C$ and ${}^{1}H$ are obtained 1 :

¹ The basic plot options for using gnuplot are plot "mydata" using 2:3 w i, "mydata" using 2:3:1 with labels