Quantum chemical calculations – Guide from the perspective of a synthethic chemist

May 2026

Dr. Tomáš Neveselý

With the generous help from

Ann-Kathrin Rückert

Iulia Turcas

Disclaimer

- This document is made by organic chemists for organic chemists and thus lot of the theoretical parts are omitted and the practical side of calculation is more at focus
- This presentation can still change as new features are added and bugs are ironed out
- This work alone is not a substitute for studying from other sources

Summary and introduction

- This document will specifically talk about DFT methods and won‘t mention Hartree-Fock or post HF methods.
- Basic type of calculations ranging from optimization and orbital properties to spectroscopy will be discussed.
- The practical side focused on software will provide information to generating inputs and reading outputs from Gaussian and sometimes Orca together with some auxiliary programs. However, once you will learn how to proceed with one, learning the second one is much easier.
- Over time this guide will include more and more Orca guides since it has many benefits over Gaussian like having much more frequent updates and thus bringing the new relevant methods earlier and being free at the same time. Also Orca contains additional methodologies like conformer search (GOAT) which is found only there.
- **Calculations are not only useful for justification a mechanism in a paper but can be used to guide your research in order to understand the underlying problems and design molecules or systems with better suited properties.**

Even a warehouse worker in a screw factory can read Vergilius in Latin. **Prof. Hrbolek**

Recommended literature – software documentation

- Gaussian 16

  Keyword list: https://gaussian.com/keywords/
  Common Gaussian errors and solutions:
  https://wongzit.github.io/gaussian-common-errors-and-solutions/#unrecognized-atomic-symbol

- Orca

  Orca forum, contains user guide and much more: https://orcaforum.kofo.mpg.de/index.php
  Little library with more examples: ***https://sites.google.com/site/orcainputlibrary/home***

- Chemcraft

  Visualization software: https://chemcraftprog.com/

- CYLview

  Pretty pictures of calculated structures for publications: https://www.cylview.org/

Recommended literature – papers

- An article summarizing what to do and what not when performing quantum chemical calculations

Angew. Chem.Int. Ed. **2022**, 61, e202205735

- A large benchmark of DFT methods from the perspective of thermochemistry

Phys. Chem. Chem. Phys., ***2017**, 19, 32184-32215*

- An overview of benchmarks of TD-DFT methods for calculations of excited states

Int. J. Quantum Chem. ***2013**, 113, 2019–2039*

- Another excellent benchmark from Head-Gordon group.

Molecular Physics 2017, 115 (19), 2315–2372

- A small guide on how to chose a method and why it **shouldn‘t** be B3LYP/6-31G.

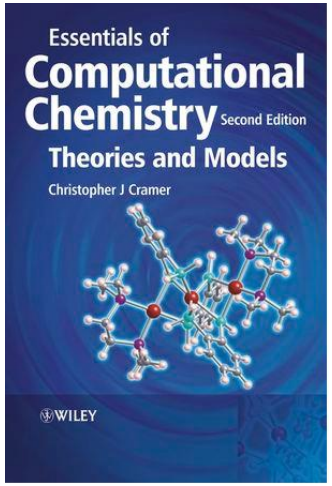
International Journal of Quantum Chemistry 2020, 120 (18).

Recommended literature – books

- Essentials of Computational Chemistry: Theories and Models, 2nd Edition

  Christopher J. Cramer

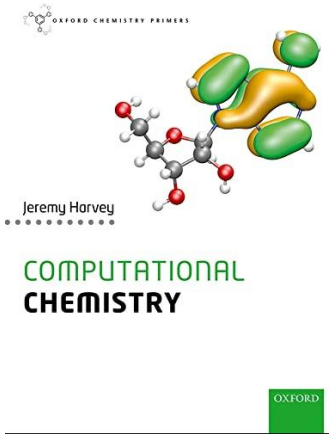
  Older book, doesn‘t contain newer development (no double hybrid functionals which are currently all the rage).



- Computational Chemistry (Oxford Chemistry Primers)

  Jeremy Harvey

  Absolute basics of computational chemistry principles. Good as a quick introduction for people untouched by calculations.



What to calculate?

Geometry

- Arguably the most important type of calculation – a starting point for all other calculations
- Transition states vs minima

Energy

- Electronic energy, enthalpy or Gibbs energy
- To compare molecules, calculate thermochemistry and related properties

Molecular properties

- Orbitals, surfaces, spin densities, ...

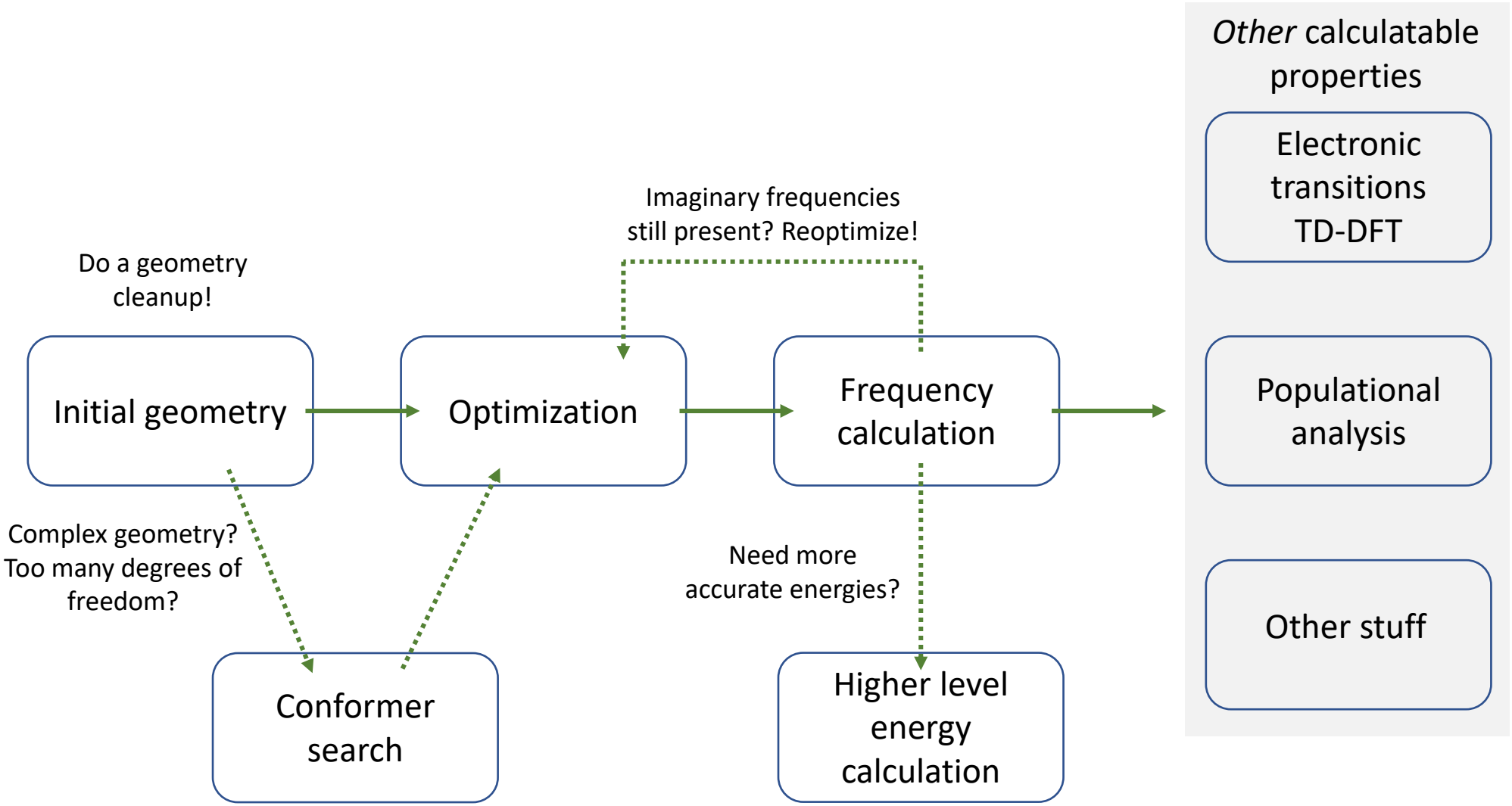
Geometry scan

- How does energy of the molecule change if we change some geometric property

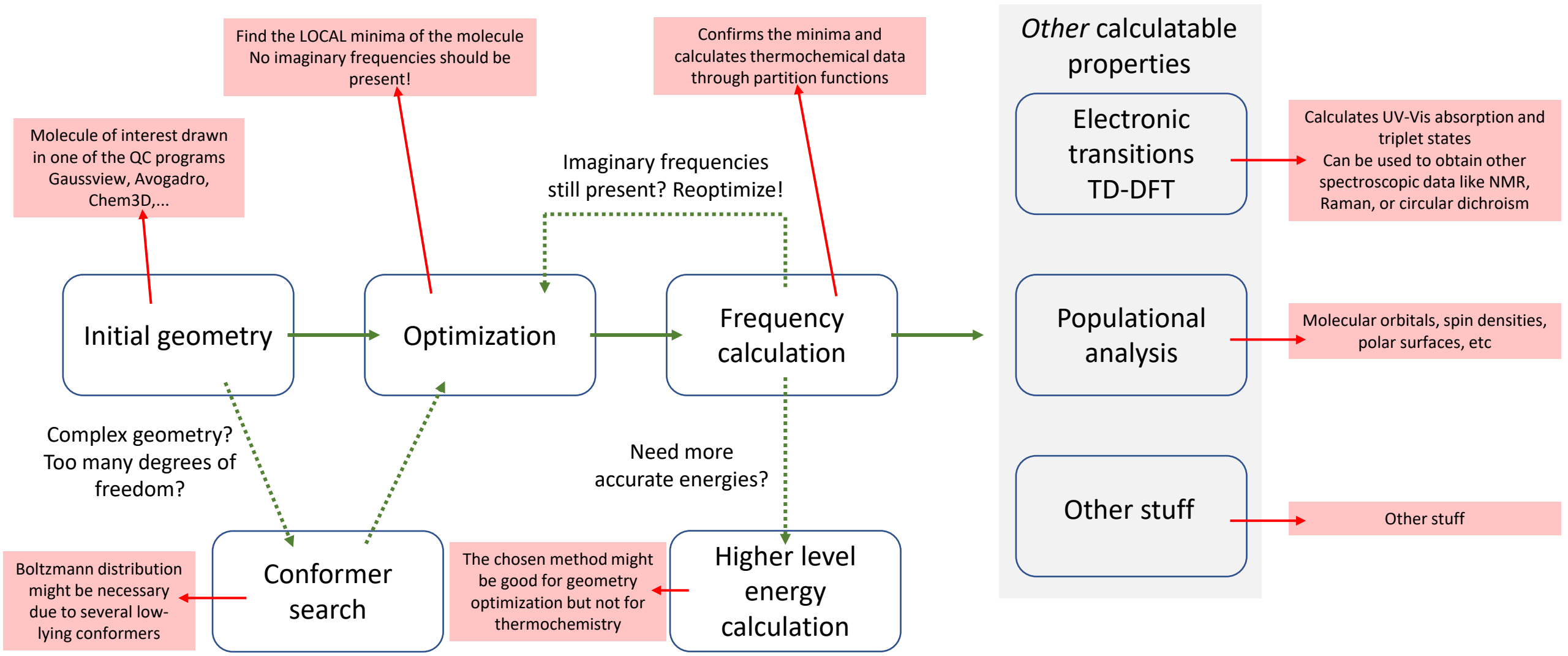
Excited states and spectra

- Absorbtion, emission in UV-Vis
- Infrared, Raman, NMR shifts, Circular dichroism, ...

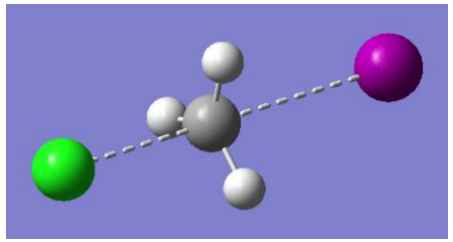
What to calculate – a typical workflow



What to calculate – a typical workflow



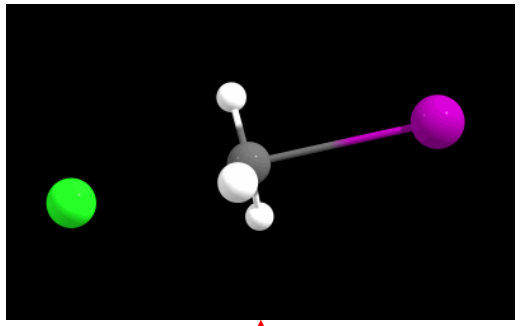
A typical workflow for the software side of calculation (Gaussian)



```
-1 1
 C                -1.22920408    0.59208431   -0.42579534
 H                -0.72109533   -0.29440701   -0.10821286
 H                -0.74611070    1.27611268   -1.09184787
 H                -2.22040440    0.79454953   -0.07732760
 Cl               -0.23548806    1.90167602    1.63990019
 I                -2.41489482   -0.97050516   -2.89053703
```

```
1 %nprocshared=10
2 %mem=5000MB
3 # opt=(calcfc,ts) tpsstpss/def2tzvp/def2tzv EmpiricalDispersion=GD3BJ
4 scrf=(solvent=acetone)
5
```

```
neveselt@chccs:~> sh god.sh SN2_TS 10 p SN2_TS
```



Draw the molecule or take coordinates from previous job (GaussView, Avogadro, Chemcraft)

Finalize the input in the text editor and check the syntax and free lines

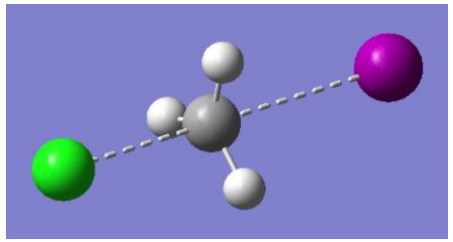
Submit the file on the cluster. Wait till its done.

Check the outcome in the text file, Chemcraft or Gaussview and process the data

Be done!

Or use the acquired data for the next input file!

A typical workflow for the software side of calculation (Gaussian)



```
-1 1
C                 -1.22920408    0.59208431   -0.42579534
H                 -0.72109533   -0.29440701   -0.10821286
H                 -0.74611070    1.27611268   -1.09184787
H                 -2.22040440    0.79454953   -0.07732760
Cl                -0.23548806    1.90167602    1.63990019
I                 -2.41489482   -0.97050516   -2.89053703
```

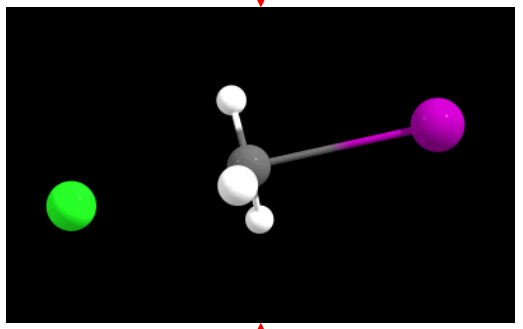
What functional and basis set to use? Do I need solvent correction? What am I calculating in the first place?
Did I forget some empty line again?

```
%nprocshared=10
%mem=5000MB
# opt=(calcfc,ts) tpsstpss/def2tzvp/def2tzv EmpiricalDispersion=GD3BJ
scrf=(solvent=acetone)

```

How many CPUs are likely needed? How much memory? How long will it take?

```
neveselt@chccs:~> sh god.sh SN2_TS 10 p SN2_TS
```

Transition state obtained; it has one imaginary frequency! *(If not, resubmit with different conditions. E.g, change geometry or use a different algorithm.)*



Now I can perform IRC calculation with this geometry!

Draw the molecule or take coordinates from previous job (GaussView, Avogadro, Chemcraft)

Finalize the input in the text editor and check the syntax and free lines

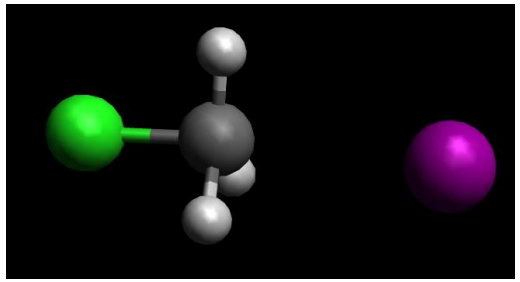
Submit the file on the cluster. Wait till its done.

Check the outcome in the text file, Chemcraft or Gaussview and process the data

Be done!

Or use the acquired data for the next input file!

A typical workflow for the software side of calculation (ORCA)



```
* xyz -1 1
  C       -8.81528        3.83608        0.16898
  I       -6.17393        4.22270       -0.64388
  H       -9.22168        4.16122       -0.74261
  H       -8.63246        4.46955        1.05253
 Cl      -11.52518        3.47905        0.88744
  H       -8.47859        2.72733        0.08370
*
```

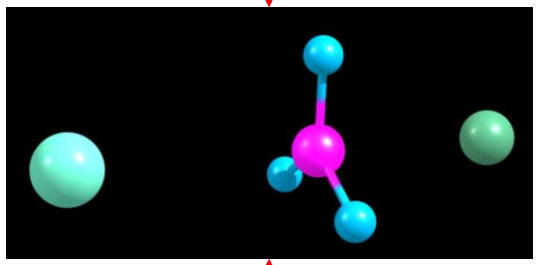
What functional and basis set to use? Do I need solvent correction? What am I calculating in the first place?
Did I forget some empty line again?

```
!wb97x-3C optts freq cpcm(acetone)
%maxcore 8000
%pal
nprocs 10
end
```

How many CPUs are likely needed? How much memory? How long will it take?

```
neveselt@chccs:~> sh god.sh SN2_TS 10 p SN2_TS
```

Transition state obtained; it has one imaginary frequency! *(If not, resubmit with different conditions. E.g, change geometry or use a different algorithm.)*



-529.12	0
157.62	5.31
166.86	17.48
167.66	17.48
934.53	1.31
935.34	1.39
1047.26	576.25
1406.06	15.62
1407.75	15.68
3239.64	5.52
3460.38	53.41
3461.05	54.42

Now I can perform IRC calculation with this geometry!

Draw the molecule or take coordinates from previous job (Chemdraw, Avogadro, Chemcraft)

Finalize the input in the text editor and check the syntax and free lines

Submit the file on the cluster. Wait till its done.

Check the outcome in the text file, Chemcraft or Avogadro and process the data

Be done!

Or use the acquired data for the next input file!

What method to chose?

Geometry optimization can be performed with a wide range of functionals fairly acurately. The main problem stems typically from presence of hydrogen bonding or other weak (Van der Waals interactions, dispersion interactions, ...). Therefore, the more complex interactions are present, the higher level DFT is necessary to perform the optimization.

What to pay attention to:

- Does the structure contain some weak non-covalent interactions?
- Is it an anion?
- Will be the optimization done in a solvent or just in a vacuum? And how much can it influence the geometry?
- Use dispersion corrected functionals or add the correction manually. Even older functionals (like B3LYP or B97) give far more accurate and reliable results with said corrections!

Suggested functionals

- To get a quick qualitative insight into a problem, **usually**, fairly low-level methods with small basis sets are sufficient. For first studies PBE0-D3BJ, TPSS-D3BJ or similar. As basis set Def2-SVP or Def2-SVPD.
- For pure functional (without Hartree-Fock portion), an approximation called resolution of identity (or sometimes density fitting) can be used. This greatly enhance speed without sacrificing accuracy. (Typical example would be TPSS-D3BJ/Def2TZVP/Def2TZV).

No matter what you calculate, there is a benchmark of DFT method which you can use as a guidance which methods perform well and which to avoid!

All of this can be somewhat „skipped“ with composite methods!

Composite methods (Orca)

Paring a DFT functional with an appropriate basis set is necessary to form a balanced method for the required job. And while most of the time we can get away with double ζ for a quick calculation to get the feeling and a larger triple or quadruple ζ for energy calculations, sometimes error stemming from basis set incompleteness or basis set superposition error (BSSE) can cause significant issues. Not to mention presence (or absence) of dispersion corrections and other factors.

Composite methods present a neat little package of all the neccesary components in one keyword!

Composite methods contain specially design functional with basis set and other correction to provide balanced method giving reliable and accurate results overcoming the shortcomings of badly chosen methods at fraction of computational costs normally connected with using high level approaches.

Examples: HF-3c, PBEh-3c, r2SCAN-3c, wB97x-3c.

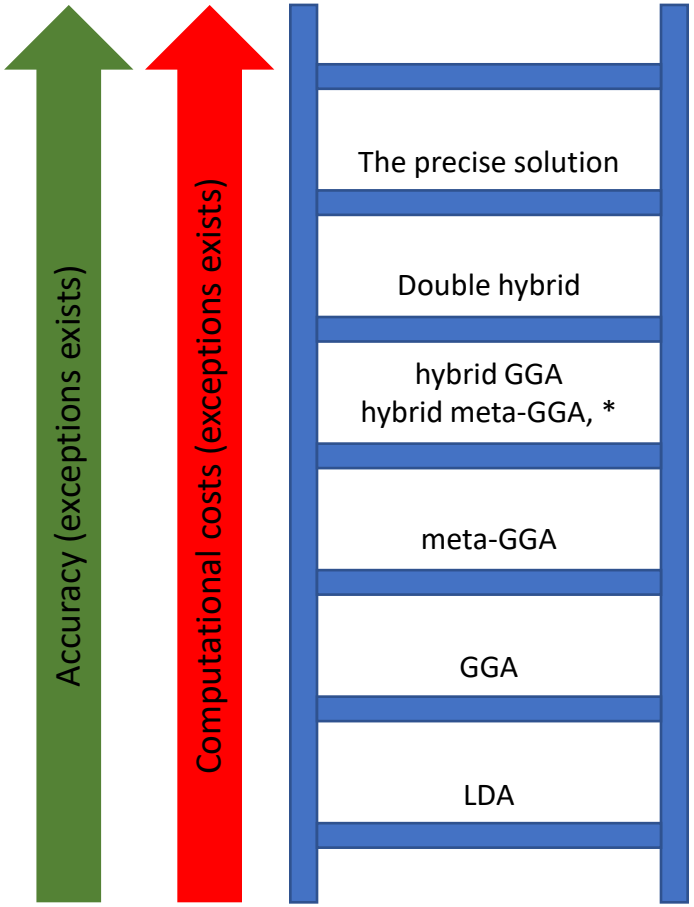
Recommended method: **wB97x-3c** Which represents especially robust method for geometry optimization, energy calculations and other. The quality of the results is comparable to high level DFT coupled with quadruple ζ basis set and yet is is much faster.

Notice, the only method keyword is **wB97x-3c**. No basis set or dispersion related keywords.

```
1  ! wB97X-3c OPT freq CPCM(Acetonitrile)
2  %maxcore 8000
3  %pal
4  nprocs 10
5  end
6  %scf
7      MaxIter 500
8  end
9
10 * xyz 0 1
11 6        4.201597000     -2.541695000     -0.449184000
```

DFT ladder

DFT functionals can be separated into several groups based their formulations. As a rule of thumb, the higher on the ladder, the higher the complexity of the functional and its precision is better. This have exceptions.



Description	Typical use case	Example
Nobody wants to share the formula	For chemist to lose their jobs	
Instead of HF these functional have incorporated Møller-Plesset correlation	Energies	PWPB95, B2PLYP, PBE0DH
Hybrid functionals add Hartree-Fock portion to the calculation	Optimization, energies, TD-DFT	PBE0, M06-2X, B3LYP,
Meta-generalized gradient approximation – Adds second derivative of the density	Optimization, TD-DFT	TPSS
Generalized gradient approximation – Adds gradient to the local density	Optimization, TD-DFT	PBE, BLYP,
Local density approximation – the granddaddy of DFT	Not used anymore, but can provide good geometries	SWVN

*Special category are range-separated hybrids where part of the energy is obtained from pure (or hybrid) functional but past a certain point and exact Hartree-Fock exchange is calculated. These methods are especially good for non-covalent interactions and excited states and sometimes energies. (LC-wPBE, wB97xD, LC-BLYP, CAM-B3LYP, ...)

Common benchmarks

To assess a quality of a particular method, there is a plethora of tests and model reactions or properties to calculate in order to find out the accuracy of the method and validate the approach.
Any property can be benchmarked: from thermochemistry, excited states to non-covalent interactions.

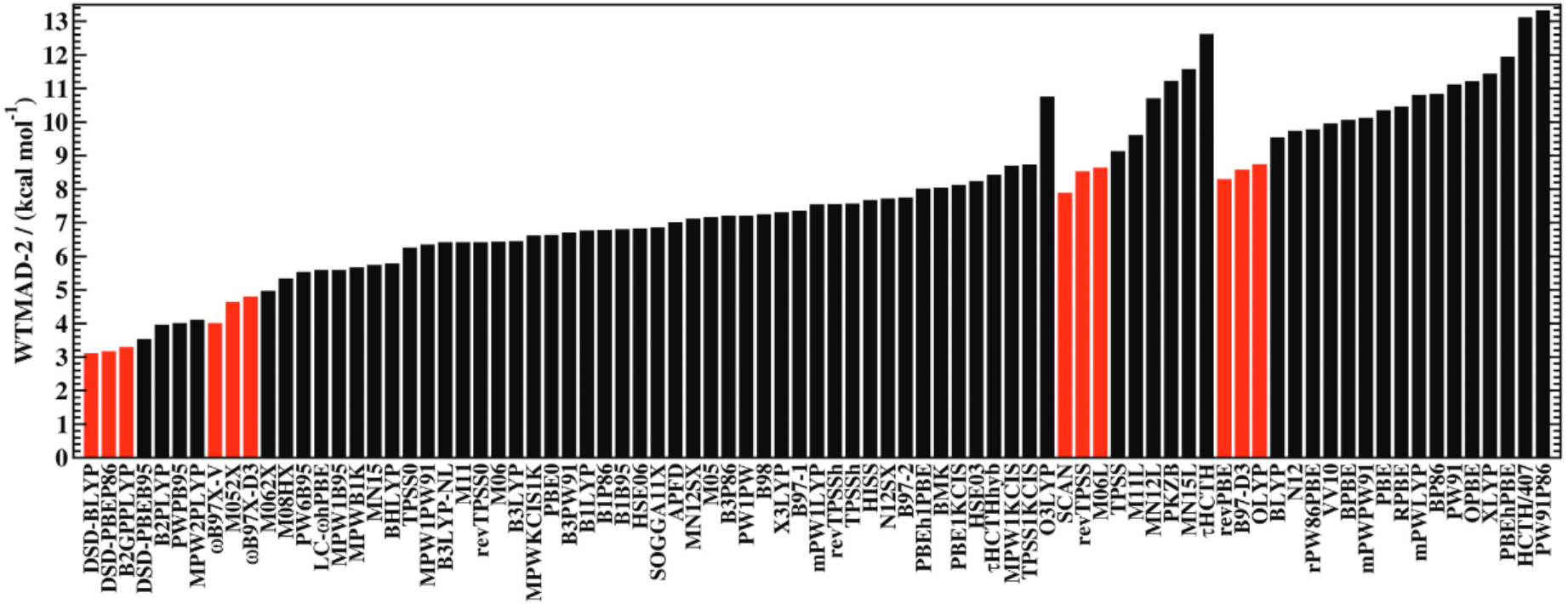


Fig. 8 Final WTMAD-2 values over the entire GMTKN55 for all assessed 83 dispersion-corrected DFAs (kcal mol^{-1}). The red bars indicate the three best approaches on their respective rung of Jacob's Ladder. The suffix "D3" was omitted in all cases, unless it is needed to avoid ambiguity.

Phys. Chem. Chem. Phys., **2017**, *19, 32184-32215*

Why B3LYP is NOT a Good Idea

1. It's OLD and Outdated
   Developed in the 90s → Better functionals exist today (ωB97X-D, PBE0, SCAN, etc.).
2. It Lacks Dispersion (van der Waals Forces)
   - ❌ Fails for noncovalent interactions (π-stacking, hydrogen bonding, etc.).
   - ✔ You *can* add Grimme's D3 correction, but why not just use a better functional?
3. Self-Interaction Error (SIE) Messes Up Energies
   - ❌ Overdelocalizes electron density → Bad for charge transfer, radicals, anions.
   - ❌ Underestimates ionization energies and electron affinities.
   - ❌ Breaks dissociation of molecules (e.g., H_2^+ dissociates into fractional charges).
4. Fails for Transition Metals & Redox Chemistry
   - ❌ Overdelocalized d-orbitals → Wrong oxidation states, bad reaction energies.
   - ❌ Poor accuracy for TM-containing reactions.
5. Underestimates Reaction Barriers
   - ❌ Activation energies are too low → Overestimates reaction speeds.
   - ⚠ This means bad kinetics predictions in catalysis & organic reactions.
6. Bad for Excited States (TD-DFT Failures)
   - ❌ Overestimates charge-transfer excitations.
   - ❌ Poor accuracy for UV-Vis spectra predictions.
7. People use it out of habit, not because it's the best.

It became a "default" choice, but benchmark studies show better options exist.

Dispersion correction

Most standard functionals do not take dispersion interactions into consideration, leading to inaccurate predictions of binding energies, molecular geometries, and non-covalent interactions. There are several schemes which can be combined with most of the common functionals to alleviate this issue.

The most prevalent schemes for dispersion correction were made by the **Grimme group:**

- The most modern scheme is D4.
- However, the older ones like D3 and D3BJ are still widely in use.

<u>For the availability check the Orca or Gaussian manual where it is shown for which DFT functionals are these schemes defined.</u>

Input:

Gaussian (has only D3 and D3BJ)

Keyword empiricaldispersion=GD3BJ

```
%nprocshared=10
%mem=6000MB
# opt=(calcfc,ts) gen scrf=(solvent=acetonitrile) empiricaldispersion=gd3bj pbe1pbe

Spiropyran opening with DEF2-SVPD

0 1
 C                 -4.67629738   -0.88738301   -0.33501194
 C                 -4.08125924    0.32650330   -0.71024193
 C                 -3.94087651   -1.79363064    0.43041836
 C                 -2.62993475   -1.53162344    0.83177954
 C                 -2.07994553   -0.32637509    0.43937657
 C                 -2.78611896    0.59887735   -0.32335866
```

Orca (has also D4)

Keyword is part of the functional

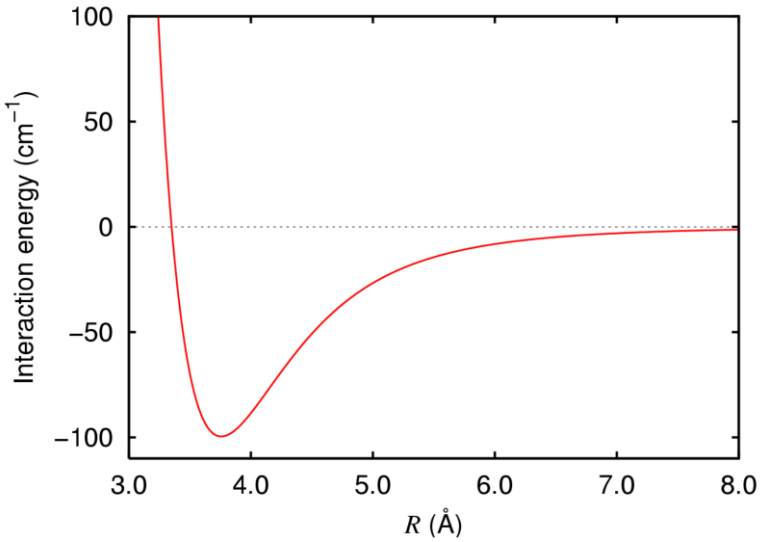
```
! wb97m-D4 OPT numFreq def2-TZVPD def2/J RijCosX CPCM(Acetonitrile)
%maxcore 8000
%pal
nprocs 10
end
```

*Some functionals contain dispersion corrections by default (like wB97xD, or wB97MV but these are more exceptions than a rule).

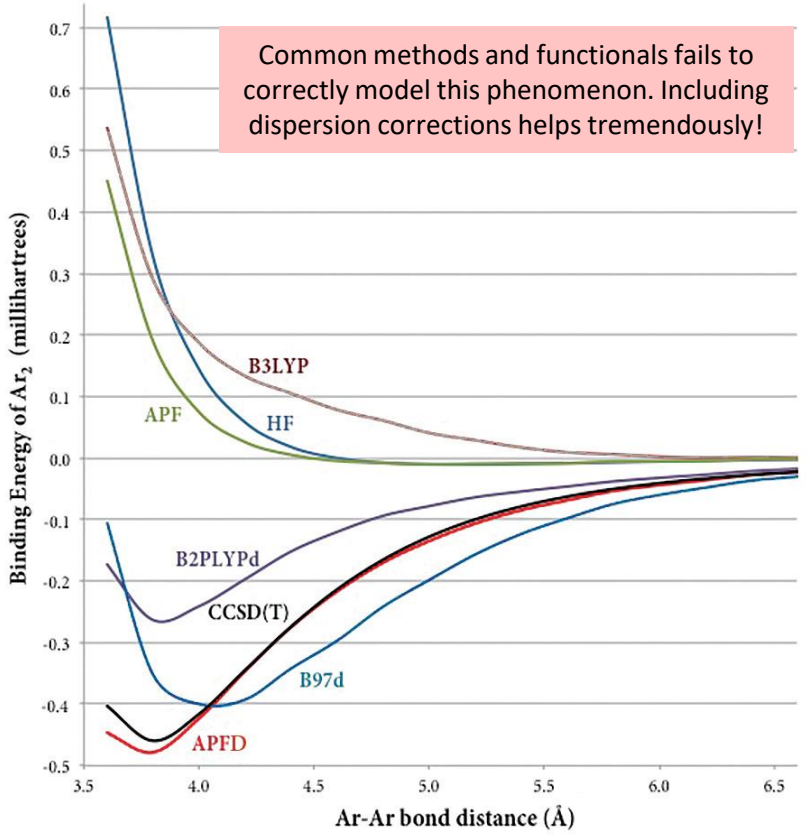
Argon dimer as a benchmark for dispersion forces

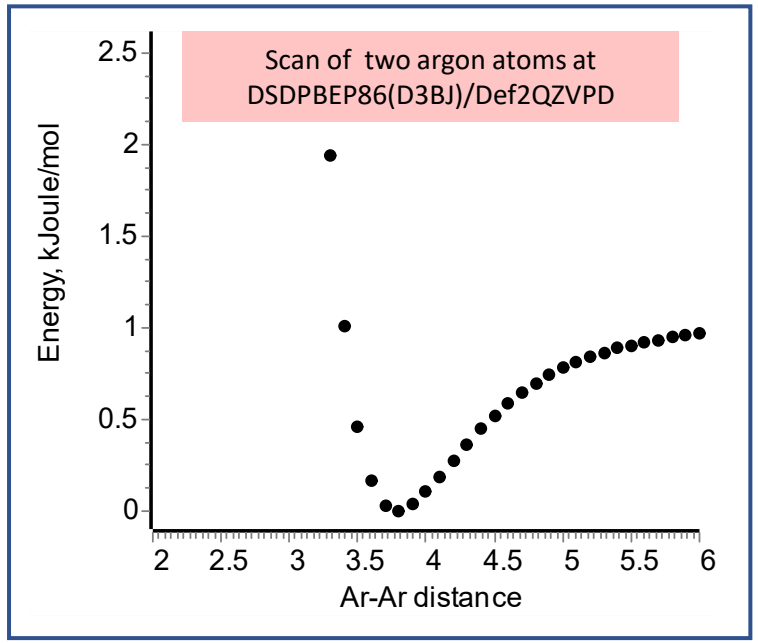
- A common way to assess the quality of weak interaction calculation is to simulate argon dimer. Noble gas even though not forming covalent bonds, they still reach an equilibrium geometry with well defined minima.

The energy of this interaction was measured to be 98.7 cm^{-1}. Or ~1.18 kJ/mol. Due to the small value and sensitivity and inherent simplicity, this became a popular testing scheme.



Common methods and functionals fails to correctly model this phenomenon. Including dispersion corrections helps tremendously!





Arguably the most common dispersion correction scheme was developed by Stefan Grimme and coworkers. The newest variant is D4, although D3 and D3BJ are still widely used.

https://expchem3.com/imagespage/cvd/argon-dimer-binding-energy/

Approximations – RI and RIJCOSX

(Orca)

To speed up the calculations there are several **approximations** which can be applied with **minimal impact to accuracy**. In general, they can be separated into two groups. Approximations applied to the Coulomb integrals (**RI**, **RI-J**) and exchange integrals (**K**, **COSX**) which are relevant for HF, or hybrid and double hybrid functionals. **RI** can be used also for MP2 based methods. Another popular approach is based on reduction of computational cost for calculation of electronic correlation (DLPNO).

It is possible to combine some approximation together for a higher acceleration of calculations. Each approximation needs an appropriate auxiliary basis set. In some scenarios certain combinations might be better than others (RIJK is faster for smaller molecules than RIJCOSX, but slower for larger ones).

Examples

- ! PBE0 def2-TZVP def2/J RIJCOSX (*approximation for Coulomb and HF Exchange of this hybrid functional*)

  Auxiliary basis set for RIJCOSX

- ! RI-B2PLYP def2-TZVP def2-TZVP/C RIJCOSX def2/J (*RI approximation for the MP2 step and RIJCOSX for the SCF*)

  Auxiliary basis set for RI or DLPNO

- ! DLPNO-B2PLYP def2-TZVP def2-TZVP/C RIJCOSX def2/J (*DLPNO approximation for the MP2 step and RIJCOSX for the SCF*)

Integration grids (Orca)

In **DFT**, the energy and density are computed using **numerical integration** over a **grid of points** in space. The grids consist of radial points (distance from the nucleus) and angular points (distribution around the nucleus).

The basic grid setting in Orca can be changed using keyword „**DefGrid#**“ where # is a number of the scheme.

- DefGrid1 = the fastest scheme with the least amount of points
- DefGrid2 = default option, balanced for speed and accuracy
- DefGrid3 = fine grid for more precise (but slower) calculations

The grid settings can be tuned more finely using specific keywords.

Convergence critera

To be added

The need for a basis set – functional alone is not enough!

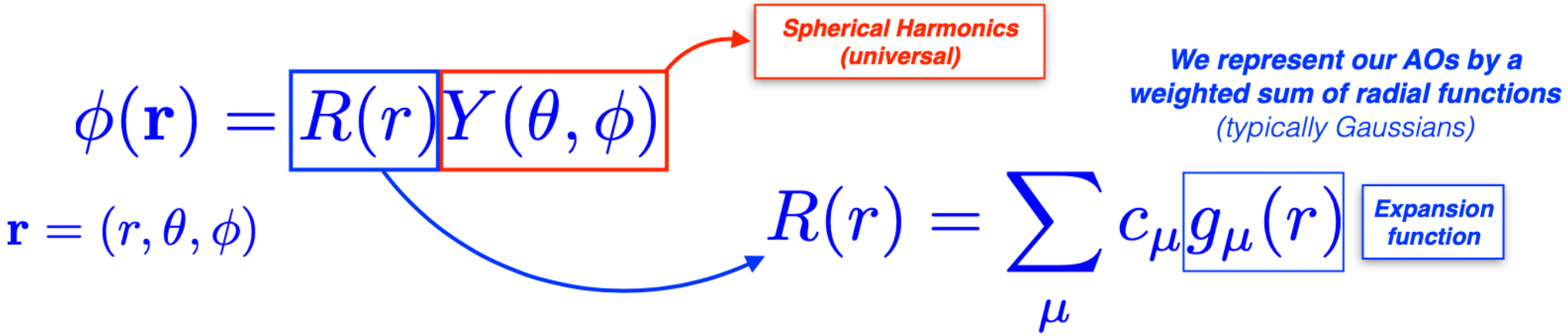
To run a successful calculation we need "method" and "space" to do it

Functional maps a number to a function. In the sense of DFT it assigns **energy** to a system with particular electron density.

Basis set is a set of orbitals which is used to describe atoms where the calculation takes place.
Sufficiently large basis set is necessary to provide adequate freedom for the functional.

Basis sets : What is a basis set ?

Reminder : Atomic orbitals (AOs) are solutions of a central-field problem (Coulomb potential created by a nucleus)



Each nucleus will carry out a set of functions describing the electrons.

The radial part of the OAs is proportional to $\mathbf{e}^{-\zeta r}$ which describes how the probability of finding an electron at a certain distance from the atomic nucleus decreases exponentially as the distance r increases. The constant ζ controls how fast this decay occurs.

Basis sets : What is a basis set ?

A basis set is a set of mathematical functions describing the atomic orbitals (1s, 2s, 2p...) of the atoms present in the studied system (carbon, hydrogen...etc).

Some basis sets are more complete (Dunning basis sets) than others (Pople basis set) in terms of the number of different atoms of the periodic table they support.

ex : bromine is not defined it Pople basis set so the energy of benzyl bromide cannot be calculated using the 6-311G* basis set.

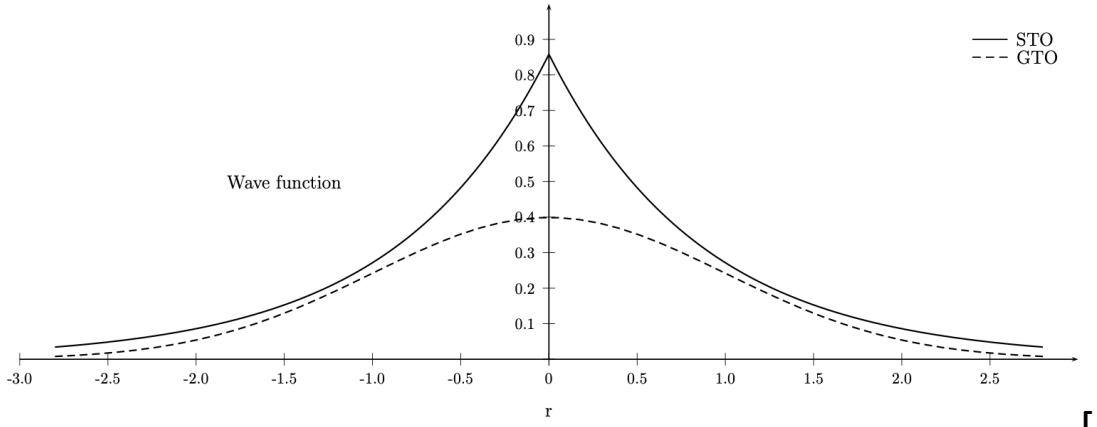
AOs are built for each atom using a set of basis functions,
then, through LCAO,
molecular orbitals (MOs) and WFs are built.

Basis sets : What is a basis set ?

Different ways to mathematically approximate the behavior of OAs (proportional to $\mathbf{e^{-\zeta r}}$) :

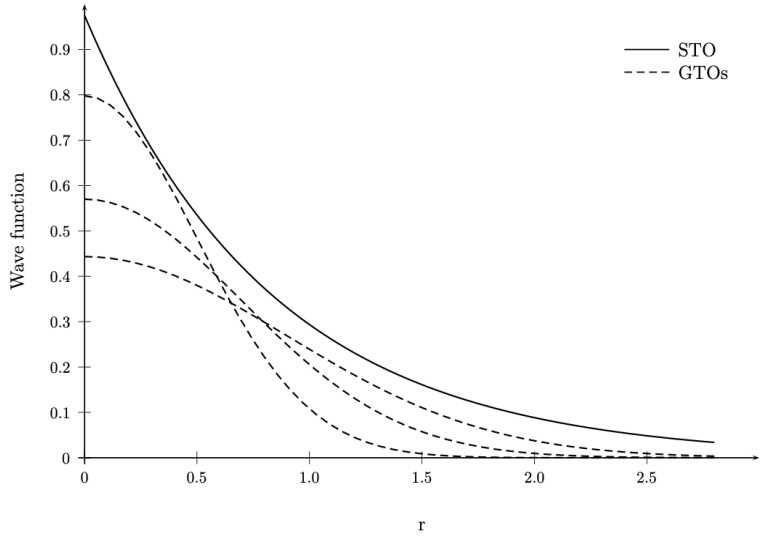
Slater-Type Orbitals (STOs) : φSTO (r)= A*exp(−ζr) with ζ : the effective nuclear charge parameter
--> Restricted use mostly due to mathematical integration difficulties

Gaussian-Type Orbitals (GTOs) : φGTO (r)= A*exp(−αr^2) with α controlling the width of the gaussian



[1]

Comparison of the shape of STO and GTO functions



[1]

Construction of a STO with 3 GTOs

[1] Hoggan, P., Ruiz Ruiz, M. B. & Ozdogan, T. Molecular Integrals over Slater-type Orbitals. From pioneers to recent progress. in 63–90 (2010).

Basis sets : Descriptors of a basis set

1. ZETA : double (DZ), triple (TZ), quadruple zeta (QZ), characterizing the increase of the number of basis functions describing the atomic orbitals of each atom. (consistent within one type of basis set)

A higher zeta number allows more flexibility in describing the electron density/WF around the atoms and can lead to more accurate results at the expense of a higher computational cost.

2. Polarization functions : Orbitals can deform; for instance, the 1s orbital of hydrogen may become slightly ovoid due to polarization. Polarization functions that describe OAs with higher quantum numbers than valence orbitals (e.g., p orbitals for hydrogen, d orbitals for second-period elements) can be added to a basis set to allow more shape deformation.

3. Diffusion functions : To describe anions (larger than neutral molecules), diffuse functions are added. These functions are of the same type as for valence orbitals but have a greater spatial expansion. Pi stacking and long-range interactions also benefit.

Basis sets : Example : 6-311+G(d,p)

The **6**-311+**G**(d,p) acronym signifies :

Atomic core orbitals (for all atoms) = sum of **6** Gaussian functions

Valence orbitals = three (311) functions, with the first being a sum of **3** Gaussians (rather close to the nucleus), while the other two consist of 1 Gaussian each (11) (further away from the nucleus).

The '**G**' indicates that the functions are gaussians.

Diffuse functions = '**+**' . '+' = diffusion for atoms heavier than hydrogen. '++' = hydrogen atoms are also included.

Polarization functions describing d and p orbitals = (d,p) or '**' (so 6-311+G**).

"=" means here "described by"

Basis sets : Considerations

Each calculation is performed at a certain level of theory, meaning a chosen method and a chosen basis set.

Different methods lead to different potential energy surfaces (ideally similar though !)
AND ALSO same method and different basis will lead to different results.

Ex : Interaction energy of benzene with Na+ :

E_{int}(M06/6-311G(d,p))= 96.7 kJ/mol

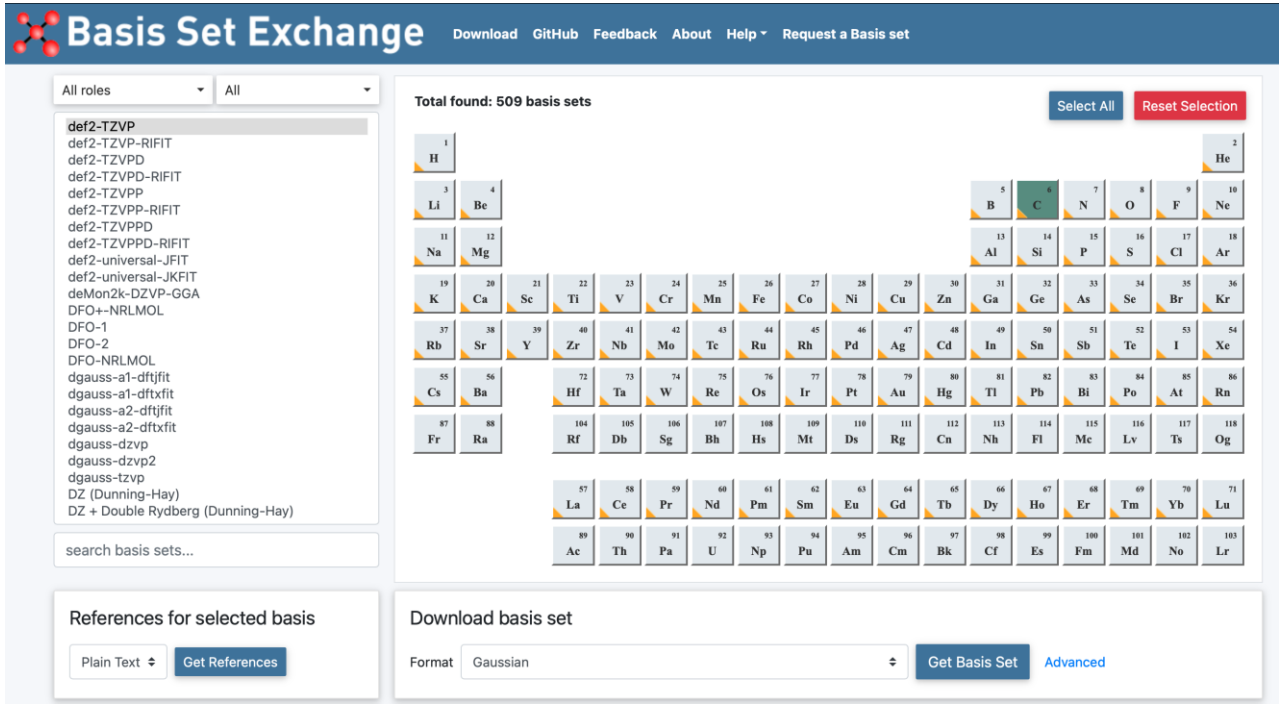
E_{int}(M06/6-31G(d,p))= 113.1 kJ/mol

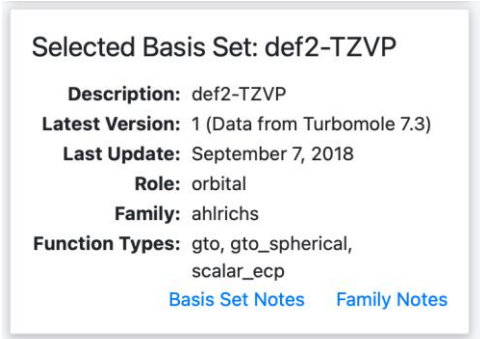
The level of theory defines unambiguously a PES and, in turn, the chemistry of a system.

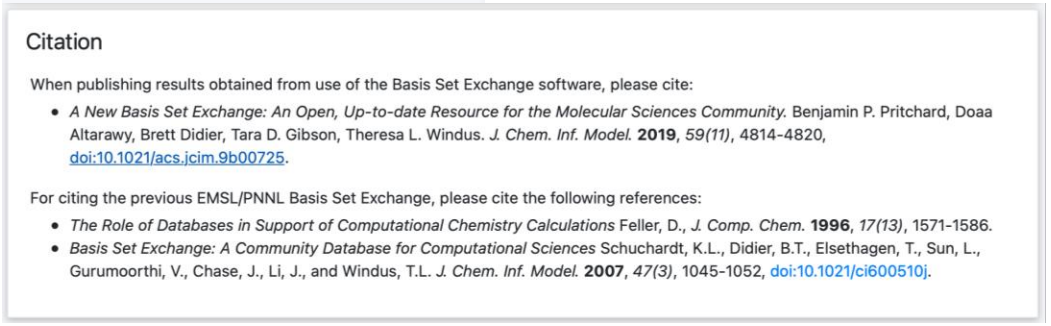
Basis sets : Recommendations

The larger the set, the better quality of the basis and the most computationally expensive its use.

Check details and atoms supported by the considered basis set on : https://www.basissetexchange.org/







Basis sets : For example : def2-TZVP and 6-311+G* basis set for carbon

```
!--------------------------------------------------------------------
!   Basis set: def2-TZVP
! Description: def2-TZVP
!        Role: orbital
!     Version: 1  (Data from Turbomole 7.3)
!--------------------------------------------------------------------


C     0
S    6   1.00
  13575.3496820              0.22245814352D-03
   2035.2333680              0.17232738252D-02
    463.22562359             0.89255715314D-02
    131.20019598             0.35727984502D-01
     42.853015891            0.11076259931
     15.584185766            0.24295627626
S    2   1.00
      6.2067138508           0.41440263448
      2.5764896527           0.23744968655
S    1   1.00
      0.57696339419          1.0000000
S    1   1.00
      0.22972831358          1.0000000
S    1   1.00
      0.95164440028D-01      1.0000000
P    4   1.00
     34.697232244            0.53333657805D-02
      7.9582622826           0.35864109092D-01
      2.3780826883           0.14215873329
      0.81433208183          0.34270471845
P    1   1.00
      0.28887547253           .46445822433
P    1   1.00
      0.10056823671           .24955789874
D    1   1.00
      1.09700000             1.0000000
D    1   1.00
      0.31800000             1.0000000
F    1   1.00
      0.76100000             1.0000000
****
```

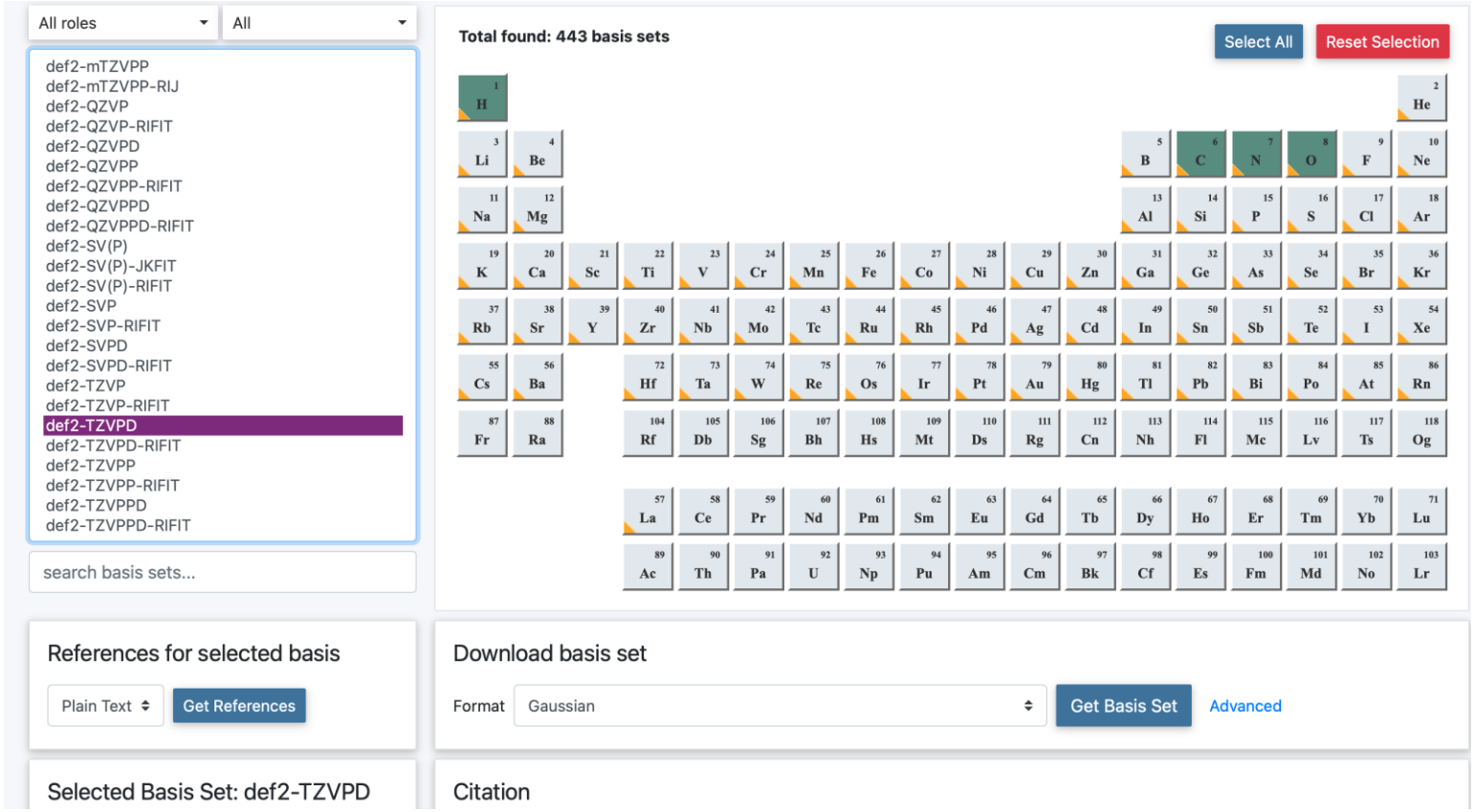
← def2-TZVP:

Multiple gaussians describe the s and p orbitals of carbon + polarization orbitals of higher angular momentum (d,f) :

6-311+G* →
Considers the 4 hybridized sp orbitals of carbon

```
!--------------------------------------------------------------------
!   Basis set: 6-311+G*
! Description: VTZ Valence Triple Zeta: 3 Funct.'s/Valence AO with
!                   diffuse+polarization on heavy atoms
!        Role: orbital
!     Version: 0  (Data from the Original Basis Set Exchange)
!--------------------------------------------------------------------


C     0
S    6   1.00
   4563.240                  0.00196665
    682.0240                 0.0152306
    154.9730                 0.0761269
     44.45530                0.2608010
     13.02900                0.6164620
      1.827730               0.2210060
SP   3   1.00
     20.96420                0.114660              0.0402487
      4.803310               0.919999              0.237594
      1.459330              -0.00303068            0.815854
SP   1   1.00
      0.4834560              1.000000              1.000000
SP   1   1.00
      0.1455850              1.000000              1.000000
SP   1   1.00
      0.0438000              1.0000000             1.0000000
D    1   1.00
      0.626                  1.000000
****
```

Basis sets : Import a basis set that isn't supported by gaussian



Choose the basis set that is not supported by Gaussian in the list on basissetexchange.org and the atoms you have in your system

Click on "get basis set" and copy/paste the parameters for each atom

gaussian specific example

Basis sets : Import a basis set that isn't supported by gaussian

```
%NProcShared=8
%mem=6000MB
# opt freq=noraman LC-wpbe EmpiricalDispersion=GD3BJ gen

EZdim

0 1
C         3.066037000     -2.254749000     -1.232980000
C         4.433172000     -2.132279000     -1.102238000
C         4.972833000     -1.600378000      0.067317000
C         4.127152000     -1.198175000      1.101314000
C         2.759881000     -1.323666000      0.959940000
C         2.213324000     -1.851962000     -0.206806000
H         6.567726000      0.055795000      1.464540000
H         7.979250000     -0.972420000      1.106834000
H         6.677499000     -1.589805000      2.151349000

H
S    3   1.00
     13.0107010              0.19682158D-01
      1.9622572              0.13796524
      0.44453796             0.47831935
S    1   1.00
      0.12194962             1.0000000
P    1   1.00
      0.8000000              1.0000000
P    1   1.00
      0.11704099050          1.0000000
****
C     0
S    5   1.00
   1238.4016938              0.54568832082D-02
    186.29004992             0.40638409211D-01
     42.251176346            0.18025593888
     11.676557932            0.46315121755
```

End of atomic coordinates/start of basis set parameters

Write "gen" to generate the basis set

Copy/paste the parameters of the basis set at the end of the atom coordinates with 1 space line in between

(ORCA usually has all the basis sets needed)

gaussian specific example

Now that we know how to select a level of theory,
we will discuss which button to press to actually run calculations.

Basics for cluster control

squeue

Checks all the jobs currently being processed by the cluster

```
neveselt@chccs:~> squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
             16728       sb1   ahrt1p ahrensmi  R 16-17:39:35      1 chcc04
             16794       sb1     EZrc turcasiu  R 7-23:46:32       1 chcc09
             16799       pb1    chg16 herwigch  R 7-23:28:23       1 chcc01
             16800       sb1    chg16 herwigch  R 7-22:43:50       1 chcc06
             16804       sb1  ZZdimer turcasiu  R 7-14:45:51       1 chcc10
             16855       sb1   EZ_ACN turcasiu  R 3-16:16:59       1 chcc06
```

sh god.sh name #cpus ϕ filename

Submits a job for calculation. File god.sh is the same for all the jobs and is reused. #CPUs should correspond to the level of jobs (typically 2 or 4 for smaller ones and 10 for large ones). Φ corresponds to the specific part of the cluster (s is for more powerful CPUs, p is for older ones). Name and filename corresponds to the name shown in the que system and to the name of the file you are submitting respectively. Files for submission needs to be .com

Example:

```
neveselt@chccs:~> sh god.sh 024_scan 10 s 024_scan
```

sh log_check.sh

Saves unfinished job file into a log-check folder. Good for checking the progress of the calculation.

scancel ######

Stops calculation of a job with particular number.

Better way to submit files (Orca example, but works for G16 as well)

Use modified files to be able to submit from any folder.

Procedure:

1. Go to the folder with input files ment for submitting (using cd command)

```
neveselt@chccs:~> cd Spiros/nitrospiro/GOAT/
```

2. Call the file for starting calculations using its full path

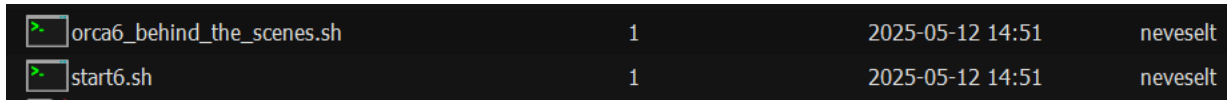
Path to your auxiliary file for starting a job.

```
neveselt@chccs:~/Spiros/nitrospiro/GOAT> sh /home/clu/neveselt/start6.sh CCC 10 s CCC
```

The name of the auxiliary file. (can differ)

3. The job is now running and its output stays in the folder from which it was submitted.

You can copy these 2 files from my directory to setup the same style submision system as well. Modify the names or the program (Orca 6 to Orca 5 or Gaussian 16) based on your needs.



Input file example – Gaussian

L1: The amount of CPU cores for the job

L3: Parameters of calculation = what is supposed to be done? Like optimization, scan, frequency. Method used (DFT functional, basis set, solvent correction, dispersion) other parameters.

L4, L6: Mandatory space between the lines containing commands on what to calculate, job description and line with charge and multiplicity! Without these, the calculation will crash.

```
%nprocshared=10
%mem=3000MB
# opt freq=noraman gen wB97xD SCRF=(Solvent=acetonitrile)

Closed spiro opt solvent

0 1
C                 -4.56409100    2.15824300   -0.35439900
C                 -3.43509300    1.77301300   -1.10463200
C                 -2.74867800    0.65804300   -0.68798000
C                 -3.12457800   -0.05312000    0.45044900
C                 -4.21924500    0.33797500    1.18603100
C                 -4.94607900    1.45731100    0.77810900
H                 -5.13009700    3.03060900   -0.66049700
H                 -3.15687600    2.33952500   -1.97823200
H                 -4.51301200   -0.20934400    2.07555900
H                 -5.80661500    1.78146400    1.35052000
C                 -2.19980000   -1.23541500    0.60600800
C                 -0.99884300   -0.77413300   -0.28721200
C                 -2.88644100   -2.46272400   -0.00003900
H                 -2.26358500   -3.35575800    0.09345400
H                 -3.81952000   -2.64571100    0.53584200
H                 -3.13008500   -2.30529400   -1.05267100
C                 -1.78552100   -1.53564900    2.03582300
H                 -2.64972600   -1.89528000    2.59856500
H                 -1.02421400   -2.31958600    2.06067700
H                 -1.39131700   -0.65312500    2.53604700
N                 -1.62629300    0.07410100   -1.26705800
C                 -0.82146300    0.80485600   -2.20850000
H                 -1.42838300    1.09170400   -3.06894700
H                 -0.38532200    1.71254500   -1.77134100
H                 -0.01587200    0.16402600   -2.56698100
C                 -0.20987300   -1.86622600   -0.92468500
H                 -0.77865800   -2.63874100   -1.42310200
C                  1.12203500   -1.86893100   -0.95174900
C                  1.87543100   -0.80837500   -0.32448600
C                  3.25361900   -0.68747200   -0.41481100
C                  1.15624200    0.14062300    0.41766600
C                  3.89052700    0.35971800    0.22422300
H                  3.83862600   -1.40326600   -0.97783500
C                  1.81329700    1.18707100    1.06092100
C                  3.18403700    1.29827600    0.96473400
H                  1.22923200    1.89934300    1.62983700
H                  3.72079600    2.10066000    1.45178100
O                 -0.16507500    0.05181700    0.57488400
N                  5.33723300    0.47830600    0.11973100
O                  5.87035000    1.40462600    0.69561500
O                  5.92599600   -0.35620400   -0.53761000
H                  1.66443300   -2.66103400   -1.45683800

H     0
S     3   1.00
     34.0613410              0.60251978D-02
      5.1235746              0.45021094D-01
      1.1646626              0.20189726
S     1   1.00
      0.32723041             1.0000000
S     1   1.00
      0.10307241             1.0000000
P     1   1.00
      0.8000000              1.0000000
P     1   1.00
      0.95774129632D-01      1.0000000
****
```

L2: The amount of memory (RAM) per CPU

L5: Job description – only for you. Not does not give information to the software.

L7: Charge of the molecule and spin multiplicity.

L8 and further: XYZ coordinates of the molecule of interest. Can also be specified as Z-matrix.

Empty space at the end of the coordinates!

If you used user specified basis set, it needs to be located after the molecular coordinates. The basis set needs to contain parameters for all the atoms involved in the calculation. At the end of the file there needs to be two empty lines!

Output file example – Gaussian – first lines

Output file in general contains results of your calculations witch specifics differing based on what was in the input

```
Entering Gaussian System, Link 0=g16
Initial command:
/usr/local/li/G16-A03/g16/l1.exe "/scr/neveselt/Gau-12064.inp" -scrdir="/scr/neveselt/"
Entering Link 1 = /usr/local/li/G16-A03/g16/l1.exe PID=     12065.

Copyright (c) 1988,1990,1992,1993,1995,1998,2003,2009,2016,
            Gaussian, Inc.  All Rights Reserved.

This is part of the Gaussian(R) 16 program.  It is based on
the Gaussian(R) 09 system (copyright 2009, Gaussian, Inc.),
the Gaussian(R) 03 system (copyright 2003, Gaussian, Inc.),
the Gaussian(R) 98 system (copyright 1998, Gaussian, Inc.),
the Gaussian(R) 94 system (copyright 1995, Gaussian, Inc.),
the Gaussian 92(TM) system (copyright 1992, Gaussian, Inc.),
the Gaussian 90(TM) system (copyright 1990, Gaussian, Inc.),
the Gaussian 88(TM) system (copyright 1988, Gaussian, Inc.),
the Gaussian 86(TM) system (copyright 1986, Carnegie Mellon
University), and the Gaussian 82(TM) system (copyright 1983,
Carnegie Mellon University). Gaussian is a federally registered
trademark of Gaussian, Inc.

This software contains proprietary and confidential information,
including trade secrets, belonging to Gaussian, Inc.

This software is provided under written license and may be
used, copied, transmitted, or stored only in accord with that
written license.

The following legend is applicable only to US Government
contracts under FAR:

                      RESTRICTED RIGHTS LEGEND

Use, reproduction and disclosure by the US Government is
subject to restrictions as set forth in subparagraphs (a)
and (c) of the Commercial Computer Software - Restricted
Rights clause in FAR 52.227-19.

Gaussian, Inc.
340 Quinnipiac St., Bldg. 40, Wallingford CT 06492
```

Previous version, law information, proper citation of the used software, etc.

```
Cite this work as:
Gaussian 16, Revision A.03,
M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria,
M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone,
G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich,
J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian,
J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young,
F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone,
T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega,
G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda,
J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai,
T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta,
F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin,
V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand,
K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar,
J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi,
J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas,
J. B. Foresman, and D. J. Fox, Gaussian, Inc., Wallingford CT, 2016.
```

Output file example – Gaussian – input of your file

Input of the calculation. Here you can check if everything was understood correctly.

```
 76  ******************************************
 77  Gaussian 16:  EM64L-G16RevA.03 25-Dec-2016
 78                2-Dec-2023
 79  ******************************************
 80  %nprocshared=10
 81  Will use up to   10 processors via shared memory.
 82  %mem=3000MB
 83  ----------------------------------------------------------------------
 84  # opt freq=noraman def2tzvp empiricaldispersion=gd3bj pbe1pbe
 85  ----------------------------------------------------------------------
 86  1/18=20,19=15,26=3,38=1/1,3;
 87  2/9=110,12=2,17=6,18=5,40=1/2;
 88  3/5=44,7=101,11=2,25=1,30=1,71=1,74=-13,124=41/1,2,3;
 89  4//1;
 90  5/5=2,38=5/2;
 91  6/7=2,8=2,9=2,10=2,28=1/1;
 92  7//1,2,3,16;
 93  1/18=20,19=15,26=3/3(2);
 94  2/9=110/2;
 95  99//99;
 96  2/9=110/2;
 97  3/5=44,7=101,11=2,25=1,30=1,71=1,74=-13,124=41/1,2,3;
 98  4/5=5,16=3,69=1/1;
 99  5/5=2,38=5/2;
100  7//1,2,3,16;
101  1/18=20,19=15,26=3/3(-5);
102  2/9=110/2;
103  6/7=2,8=2,9=2,10=2,19=2,28=1/1;
104  99/9=1/99;
105  ----------------
106  Closed spiro opt
```

```
106  Closed spiro opt
107  ----------------
108  Symbolic Z-matrix:
109  Charge =  0 Multiplicity = 1
110  C                    -5.23842   0.92242  -2.22791
111  C                    -4.01501   0.95696  -1.52612
112  C                    -3.63421   2.16083  -0.93622
113  C                    -4.40342   3.26316  -1.02645
114  C                    -5.6164    3.28406  -1.70785
115  C                    -6.0406    2.08816  -2.3195
116  H                    -5.56174   0.01432  -2.69236
117  H                    -3.39849   0.08581  -1.44926
118  H                    -6.20528   4.17561  -1.76501
119  H                    -6.96766   2.05939  -2.853
120  C                    -3.68749   4.37964  -0.27709
121  C                    -2.2505    3.8939   -0.42267
122  C                    -4.13607   4.40471   1.19592
123  H                    -3.55894   5.12871   1.73224
124  H                    -5.17264   4.66478   1.2487
125  H                    -3.98831   3.43806   1.63026
126  C                    -3.91021   5.78937  -0.85557
127  H                    -4.94628   6.04601  -0.78077
128  H                    -3.32676   6.49722  -0.30472
129  H                    -3.61233   5.80439  -1.88317
130  N                    -2.39815   2.4524   -0.15335
131  C                    -1.21837   1.67162  -0.55257
132  H                    -1.4041    0.63188  -0.38127
133  H                    -1.01946   1.83194  -1.59163
134  H                    -0.37278   1.98271   0.02459
135  C                    -1.25351   4.60959   0.49367
136  H                    -1.50059   5.50296   1.02823
137  C                    -0.05087   4.04305   0.56398
138  C                     0.3093    3.19357  -0.66495
139  C                     1.46924   2.41183  -0.69505
140  C                    -0.55749   3.23987  -1.7903
141  C                     1.78469   1.68172  -1.84944
142  H                     2.11342   2.37276   0.15842
143  C                    -0.2255    2.51585  -2.94282
144  C                     0.94252   1.73788  -2.97157
145  H                    -0.86347   2.55459  -3.80096
146  H                     1.19201   1.18443  -3.85268
147  O                    -1.78025   4.02653  -1.75317
148  N                     3.00067   0.85652  -1.88575
149  O                     3.27159   0.23309  -2.87078
150  O                     3.71976   0.8081   -0.93029
151  H                     0.61467   4.18501   1.38969
```

Output file example – Gaussian – Results of optimization

```
13665          Item               Value     Threshold  Converged?
13666  Maximum Force            0.000011     0.000450     YES
13667  RMS     Force            0.000002     0.000300     YES
13668  Maximum Displacement     0.001252     0.001800     YES
13669  RMS     Displacement     0.000244     0.001200     YES
13670  Predicted change in Energy=-2.938925D-09
13671  Optimization completed.
13672     -- Stationary point found.
```

If the molecule was successfully optimized, and converged to the final geometry, all the criteria will be fulfilled.

```
14238  ****************************************************************************
14239
14240              Population analysis using the SCF density.
14241
14242  ****************************************************************************
14243
```

Information about orbital energies of the optimized molecule can be found in the output.

HOMO and LUMO in **Hartree**.

```
14317  The electronic state is 1-A.
14318  Alpha  occ. eigenvalues --  -19.26236 -19.23160 -19.23143 -14.61765 -14.41679
14319  Alpha  occ. eigenvalues --  -10.34357 -10.31749 -10.28381 -10.26424 -10.26139
14320  Alpha  occ. eigenvalues --  -10.26003 -10.25929 -10.25896 -10.25759 -10.25714
14321  Alpha  occ. eigenvalues --  -10.25088 -10.24786 -10.23099 -10.23017 -10.22879
14322  Alpha  occ. eigenvalues --  -10.22751 -10.22568 -10.22167 -10.21993  -1.26771
14323  Alpha  occ. eigenvalues --   -1.13511  -1.08757  -1.01030  -0.92550  -0.88748
14324  Alpha  occ. eigenvalues --   -0.87251  -0.85927  -0.82864  -0.79822  -0.77661
14325  Alpha  occ. eigenvalues --   -0.76972  -0.75648  -0.72821  -0.71174  -0.68501
14326  Alpha  occ. eigenvalues --   -0.67428  -0.65698  -0.63930  -0.61751  -0.59341
14327  Alpha  occ. eigenvalues --   -0.58591  -0.58073  -0.56035  -0.55794  -0.54490
14328  Alpha  occ. eigenvalues --   -0.52923  -0.52605  -0.51846  -0.49845  -0.48963
14329  Alpha  occ. eigenvalues --   -0.47932  -0.47348  -0.46820  -0.46040  -0.45736
14330  Alpha  occ. eigenvalues --   -0.45424  -0.45017  -0.44499  -0.43720  -0.42656
14331  Alpha  occ. eigenvalues --   -0.41938  -0.41199  -0.40583  -0.39489  -0.39106
14332  Alpha  occ. eigenvalues --   -0.38679  -0.37763  -0.36813  -0.36182  -0.35169
14333  Alpha  occ. eigenvalues --   -0.34603  -0.33761  -0.32682  -0.32498  -0.31049
14334  Alpha  occ. eigenvalues --   -0.30811  -0.28207  -0.26849  -0.25208  -0.22295
14335  Alpha virt. eigenvalues --   -0.08052  -0.05803  -0.01156  -0.00635   0.00358
14336  Alpha virt. eigenvalues --    0.03241   0.03928   0.05174   0.05463   0.06304
14337  Alpha virt. eigenvalues --    0.06397   0.07484   0.08319   0.08795   0.09412
14338  Alpha virt. eigenvalues --    0.09925   0.10594   0.10876   0.11121   0.11670
```

Input file example – Orca

L1: Contains the calculation specific keywords like method, basis set, approximations or solvation model. Besides that there is also specified what should be calculated (optimization, frequency, excited states, etc.)

L10: The source of the coordinates. Could by changed for external XYZ file by specifying its name or can be internal like in this case. This is followed by charge and multplicity of the system.

L11+: XYZ coordinates of the molecule of interest.

Orca is not as sensitive to missed spaces or lines or even placement of the keywords like Gaussian!

```
1  ! wb97x-D4 OPT Freq RIJCOSX def2-TZVP def2/J def2-TZVP/C TightSCF CPCM(Acetonitrile)
2  %maxcore 8000
3  %pal
4  nprocs 10
5  end
6  %scf
7      MaxIter 500
8  end
9
10 * xyz 0 1
11 6        -4.642144000     -2.085236000      0.322460000
12 6        -3.552236000     -1.706334000      1.105899000
13 6        -2.803250000     -0.619305000      0.689511000
14 6        -3.117416000      0.069411000     -0.480911000
15 6        -4.193070000     -0.312756000     -1.248298000
16 6        -4.964359000     -1.403933000     -0.840803000
17 1        -5.240348000     -2.934610000      0.628554000
18 1        -3.303721000     -2.254932000      2.004818000
19 1        -4.438680000      0.221265000     -2.158928000
20 1        -5.809950000     -1.720641000     -1.437059000
21 6        -2.152970000      1.222926000     -0.638894000
22 6        -1.000668000      0.754142000      0.318422000
23 6        -2.827625000      2.497392000     -0.109555000
24 1        -2.171289000      3.363680000     -0.208304000
25 1        -3.727105000      2.688110000     -0.695547000
26 1        -3.120678000      2.394228000      0.936318000
27 6        -1.673556000      1.459733000     -2.065557000
28 1        -2.500575000      1.836823000     -2.669148000
29 1        -0.879700000      2.209811000     -2.082412000
30 1        -1.299786000      0.547929000     -2.526104000
31 7        -1.689721000     -0.046339000      1.301699000
32 6        -0.927904000     -0.793083000      2.276382000
33 1        -1.566100000     -1.058601000      3.118616000
34 1        -0.503735000     -1.712202000      1.855273000
35 1        -0.117711000     -0.171426000      2.654757000
36 6        -0.199449000      1.848481000      0.948787000
37 1        -0.764209000      2.624348000      1.444558000
38 6         1.128364000      1.858496000      0.964709000
39 6         1.884053000      0.787079000      0.341723000
40 6         3.263334000      0.690280000      0.405066000
41 6         1.164646000     -0.193616000     -0.353623000
```

L2: The amount of memory (RAM) per CPU

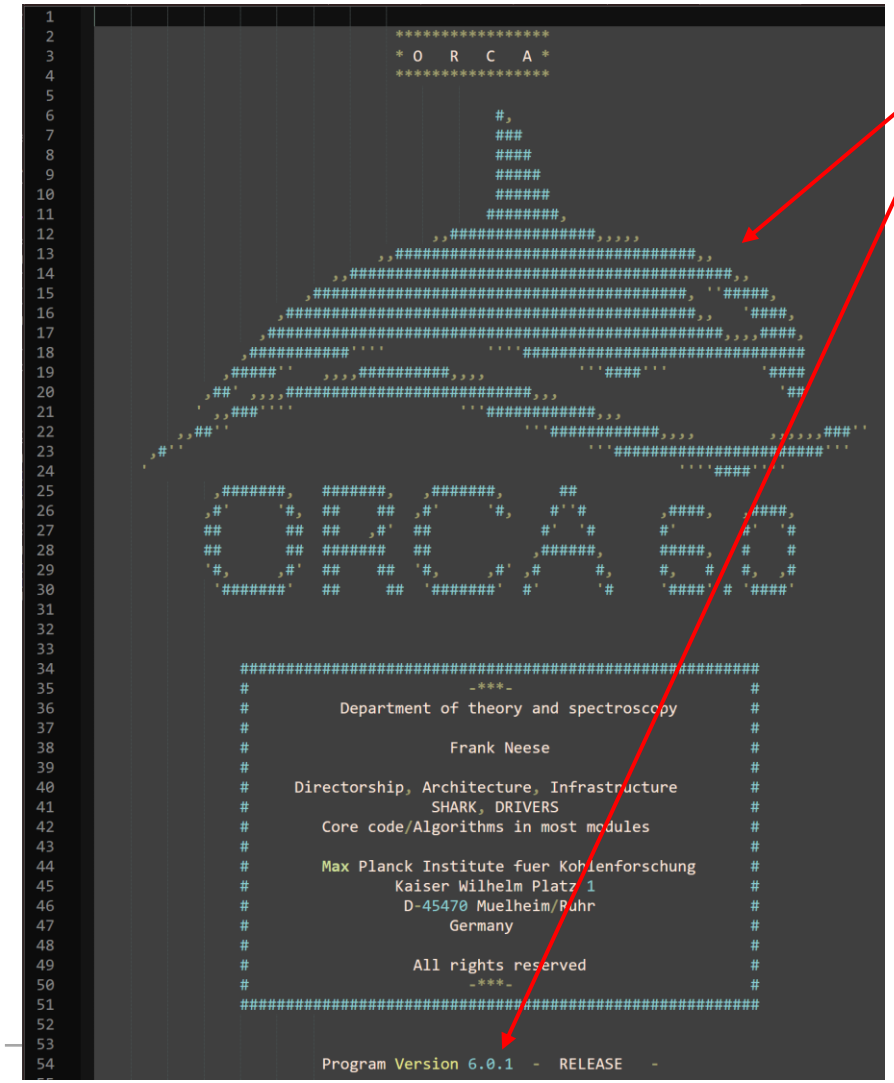
L3-5: The number of CPU cores used for the job. Each specific line starting with „%“ must be terminated with the keyword „end“.

L6+: All other specific parameters for the calculation should be here. It can be anything from the grid size to freezing cooridnates or parameters for scan.

Continuation of the coordinates.

Output file example – Orca 6 – first lines

Output file in general contains results of your calculations witch specifics differing based on what was in the input file.



Really big Orca, version of the software and so on. Underneath is a list of creators and what part of code they are responsible for.

```
With contributions from (in alphabetic order):
[Max-Planck-Institut fuer Kohlenforschung]
  Daniel Aravena         : Magnetic Suceptibility
  Michael Atanasov       : Ab Initio Ligand Field Theory (pilot matlab implementation)
  Alexander A. Auer      : GIAO ZORA, VPT2 properties, NMR spectrum
  Ute Becker             : All parallelization in ORCA, NUMFREQ, NUMCALC
  Giovanni Bistoni       : ED, misc. LED, open-shell LED, HFLD
  Martin Brehm           : Molecular dynamics
  Dmytro Bykov           : pre 5.0 version of the SCF Hessian
  Marcos Casanova-Páez   : Triplet and SCS-CIS(D). UHF-(DLPNO)-IP/EA/STEOM-CCSD. UHF-CVS-IP/STEOM-CCSD
  Vijay G. Chilkuri      : MRCI spin determinant printing, contributions to CSF-ICE
  Pauline Colinet        : FMM embedding
  Dipayan Datta          : RHF DLPNO-CCSD density
  Achintya Kumar Dutta   : EOM-CC, STEOM-CC
  Nicolas Foglia         : Exact transition moments, OPA infrastructure, MCD improvements
  Dmitry Ganyushin       : Spin-Orbit,Spin-Spin,Magnetic field MRCI
  Miquel Garcia-Rates    : C-PCM and meta-GGA Hessian, CCSD/C-PCM, Gaussian charge scheme
  Tiago L. C. Gouveia    : GS-ROHF, GS-ROCIS
  Yang Guo               : DLPNO-NEVPT2, F12-NEVPT2, CIM, IAO-localization
  Andreas Hansen         : Spin unrestricted coupled pair/coupled cluster methods
  Ingolf Harden          : AUTO-CI MPn and infrastructure
  Benjamin Helmich-Paris : MC-RPA, TRAH-(SCF,CASSCF), AVAS, COSX integrals, SCF dyn. polar.
  Lee Huntington         : MR-EOM, pCC
  Robert Izsak           : Overlap fitted RIJCOSX, COSX-SCS-MP3, EOM
  Riya Kayal             : Wick's Theorem for AUTO-CI, AUTO-CI UHF-CCSDT
  Emily Kempfer          : AUTO-CI, RHF CISDT and CCSDT
  Christian Kollmar      : KDIIS, OOCD, Brueckner-CCSD(T), CCSD density, CASPT2, CASPT2-K, improved NEVPT2
  Axel Koslowski         : Symmetry handling
  Simone Kossmann        : meta-GGA functionals, TD-DFT gradient, OOMP2, (MP2 Hessian; deprecated post 5.0)
  Lucas Lang             : DCDCAS
  Marvin Lechner         : AUTO-CI (C++ implementation), FIC-MRCC
  Spencer Leger          : CASSCF response
  Dagmar Lenk            : GEPOL surface, SMD, ORCA-2-JSON
  Dimitrios Liakos       : Extrapolation schemes; Compound Job, initial MDCI parallelization
  Dimitrios Manganas     : Further ROCIS development; embedding schemes. LFT, Crystal Embedding
  Dimitrios Pantazis     : SARC Basis sets
  Anastasios Papadopoulos: AUTO-CI, single reference methods and gradients
  Taras Petrenko         : pre 6.0 DFT Hessian and TD-DFT gradient, (ASA, deprecated), ECA, 1-Electron XAS/XES,
  Peter Pinski           : DLPNO-MP2, DLPNO-MP2 Gradient
  Christoph Reimann      : Effective Core Potentials
  Marius Retegan         : Local ZFS, SOC
  Christoph Riplinger    : Optimizer, TS searches, QM/MM, DLPNO-CCSD(T), (RO)-DLPNO pert. Triples
  Michael Roemelt        : Original ROCIS implementation
  Masaaki Saitow         : Open-shell DLPNO-CCSD energy and density
  Barbara Sandhoefer     : DKH picture change effects
  Kantharuban Sivalingam : CASSCF convergence/infrastructure, NEVPT2 and variants, FIC-MRCI
  Bernardo de Souza      : ESD, SOC TD-DFT
  Georgi L. Stoychev     : AutoAux, RI-MP2 NMR, DLPNO-MP2 response, X2C
  Van Anh Tran           : RI-MP2 g-tensors
  Willem Van den Heuvel  : Paramagnetic NMR
  Zikuan Wang            : NOTCH, Electric field optimization
  Frank Wennmohs         : Technical directorship and infrastructure
```

Output file example – Orca 6 – input of your file

Input of the calculation. Here you can check if everything was understood correctly.

```
212 ================================================================================
213                                        INPUT FILE
214 ================================================================================
215 NAME = TCC.inp
216 |  1> ! wB97X-3c OPT freq CPCM(Acetonitrile)
217 |  2> %maxcore 8000
218 |  3> %pal
219 |  4> nprocs 10
220 |  5> end
221 |  6> %scf
222 |  7>   MaxIter 500
223 |  8> end
224 |  9>
225 | 10> * xyz 0 1
226 | 11> 6        6.128163000      0.318177000     -0.403010000
227 | 12> 6        5.126949000      1.241683000     -0.129216000
228 | 13> 6        3.839228000      0.751568000     -0.007416000
229 | 14> 6        3.534617000     -0.584184000     -0.160091000
230 | 15> 6        4.533709000     -1.496808000     -0.427381000
231 | 16> 6        5.839536000     -1.033842000     -0.549189000
232 | 17> 1        7.148324000      0.663143000     -0.509321000
233 | 18> 1        5.362452000      2.292255000     -0.034440000
234 | 19> 1        4.310179000     -2.550927000     -0.535708000
235 | 20> 1        6.639063000     -1.731363000     -0.761829000
236 | 21> 6        2.060207000     -0.782664000      0.065138000
237 | 22> 6        1.567282000      0.649938000      0.197417000
238 | 23> 6        1.846655000     -1.523746000      1.401118000
239 | 24> 1        0.787161000     -1.695264000      1.568566000
240 | 25> 1        2.369910000     -2.479549000      1.360553000
241 | 26> 1        2.259339000     -0.947383000      2.231231000
242 | 27> 6        1.446321000     -1.485782000     -1.150566000
243 | 28> 1        1.871554000     -2.486936000     -1.235056000
244 | 29> 1        0.374367000     -1.575152000     -1.037942000
245 | 30> 1        1.680564000     -0.937143000     -2.064911000
246 | 31> 7        2.641524000      1.451408000      0.241595000
```

```
199 ================================================================================
200                                        WARNINGS
201                                Please study these warnings very carefully!
202 ================================================================================
203
204 WARNING: Frozen-Core approximation turned on, but there are no core electrons.
205   ===> : Turning off frozen-core approximation
206
207
208 WARNING: Geometry Optimization
209   ===> : Switching off AutoStart
210          For restart on a previous wavefunction, please use MOREAD
211
```

Output file example – Orca 6 – Energies from frequency calculations

Different thermochemical data of interest

```
51657  ------------
51658  INNER ENERGY
51659  ------------
51660
51661  The inner energy is: U= E(el) + E(ZPE) + E(vib) + E(rot) + E(trans)
51662      E(el)   - is the total energy from the electronic structure calculation
51663                = E(kin-el) + E(nuc-el) + E(el-el) + E(nuc-nuc)
51664      E(ZPE)  - the the zero temperature vibrational energy from the frequency calculation
51665      E(vib)  - the the finite temperature correction to E(ZPE) due to population
51666                of excited vibrational states
51667      E(rot)  - is the rotational thermal energy
51668      E(trans)- is the translational thermal energy
51669
51670  Summary of contributions to the inner energy U:
51671  Electronic energy                ...   -187.06618745 Eh
51672  Zero point energy                ...      0.34401685 Eh     215.87 kcal/mol
51673  Thermal vibrational correction   ...      0.01675158 Eh      10.51 kcal/mol
51674  Thermal rotational correction    ...      0.00141627 Eh       0.89 kcal/mol
51675  Thermal translational correction ...      0.00141627 Eh       0.89 kcal/mol
51676  ------------------------------------------------------------------------
51677  Total thermal energy                    -186.70258648 Eh
51678
51679
51680  Summary of corrections to the electronic energy:
51681  (perhaps to be used in another calculation)
51682  Total thermal correction                 0.01958412 Eh      12.29 kcal/mol
51683  Non-thermal (ZPE) correction             0.34401685 Eh     215.87 kcal/mol
51684  ------------------------------------------------------------------------
51685  Total correction                         0.36360097 Eh     228.16 kcal/mol
51686
51687
51688  --------
51689  ENTHALPY
51690  --------
51691
51692  The enthalpy is H = U + kB*T
51693                kB is Boltzmann's constant
51694  Total thermal energy             ...   -186.70258648 Eh
51695  Thermal Enthalpy correction      ...      0.00094421 Eh       0.59 kcal/mol
51696  ------------------------------------------------------------------------
51697  Total Enthalpy                   ...   -186.70164227 Eh
51698
```

```
51756  -------------------
51757  GIBBS FREE ENERGY
51758  -------------------
51759
51760  The Gibbs free energy is G = H - T*S
51761
51762  Total enthalpy                   ...   -186.70164227 Eh
51763  Total entropy correction         ...     -0.06591574 Eh    -41.36 kcal/mol
51764  ------------------------------------------------------------------------
51765  Final Gibbs free energy          ...   -186.76755801 Eh
51766
51767  For completeness - the Gibbs free energy minus the electronic energy
51768  G-E(el)                          ...      0.29862944 Eh    187.39 kcal/mol
51769
```

Output file example – Orca 6 – how long did everythink take

```
51890  Timings for individual modules:
51891
51892  Sum of individual times         ...    20911.090 sec (= 348.518 min)
51893  Startup calculation             ...      240.111 sec (=   4.002 min)   1.1 %
51894  SCF iterations                  ...    13726.870 sec (= 228.781 min)  65.6 %
51895  Property integrals              ...      769.413 sec (=  12.824 min)   3.7 %
51896  SCF Response                    ...      780.265 sec (=  13.004 min)   3.7 %
51897  Property calculations           ...     1542.660 sec (=  25.711 min)   7.4 %
51898  SCF Gradient evaluation         ...     3850.432 sec (=  64.174 min)  18.4 %
51899  Geometry relaxation             ...        1.339 sec (=   0.022 min)   0.0 %
51900                          ****ORCA TERMINATED NORMALLY****
51901  TOTAL RUN TIME: 0 days 5 hours 48 minutes 47 seconds 78 msec
51902
```

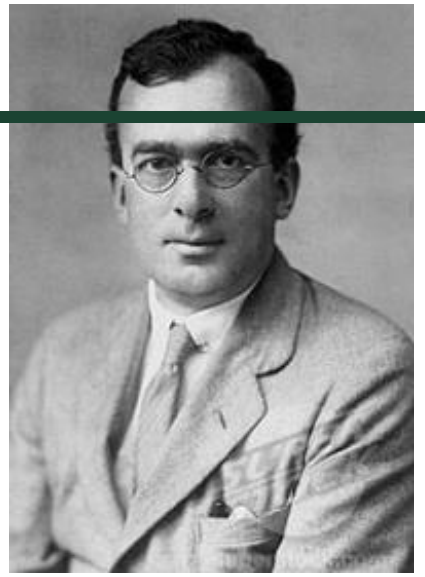
Calculation ended without an error

What is Hartree?

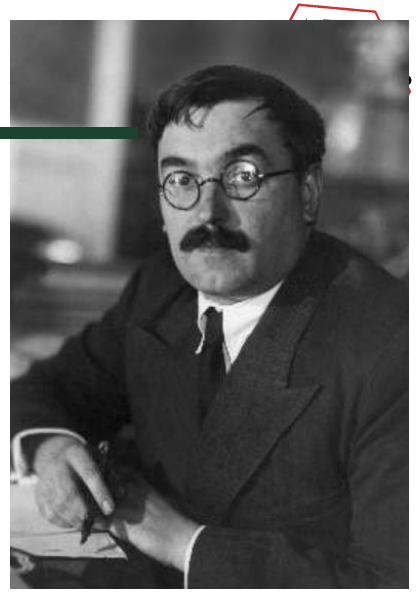
- System of units designed to simplify calculations on atomic scale
- Values of several constants were set as equal to unity

Symbol and Name	Quantity (dimensions)	Atomic units	SI units
$\hbar$, reduced Planck constant	action (ML^2T^{-1})	1	$1.054571817...\times10^{-34}$ J·s
e , elementary charge	charge (Q)	1	$1.602176634\times10^{-19}$ C
m_e , electron rest mass	mass (M)	1	$9.1093837139(28)\times10^{-31}$ kg
$4\pi\epsilon_0$, permittivity	permittivity ($Q^2W^{-1}L^{-1}$)	1	$1.11265005620(17)\times10^{-10}$ F·m^{-1}

E_h=2625.6kJ/mol (27.2eV per particle)

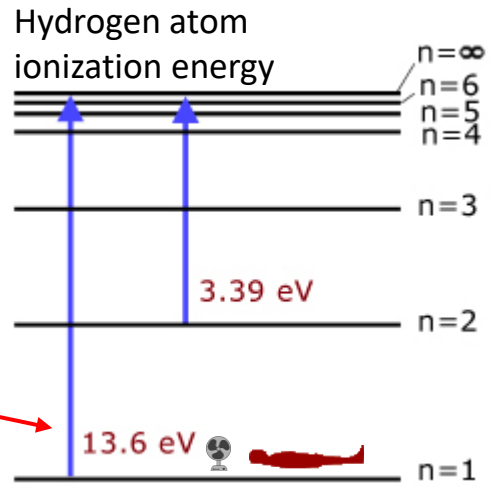


Douglas Hartree



Владимир Фок

Not exactly double due to relativistic effects, but close!



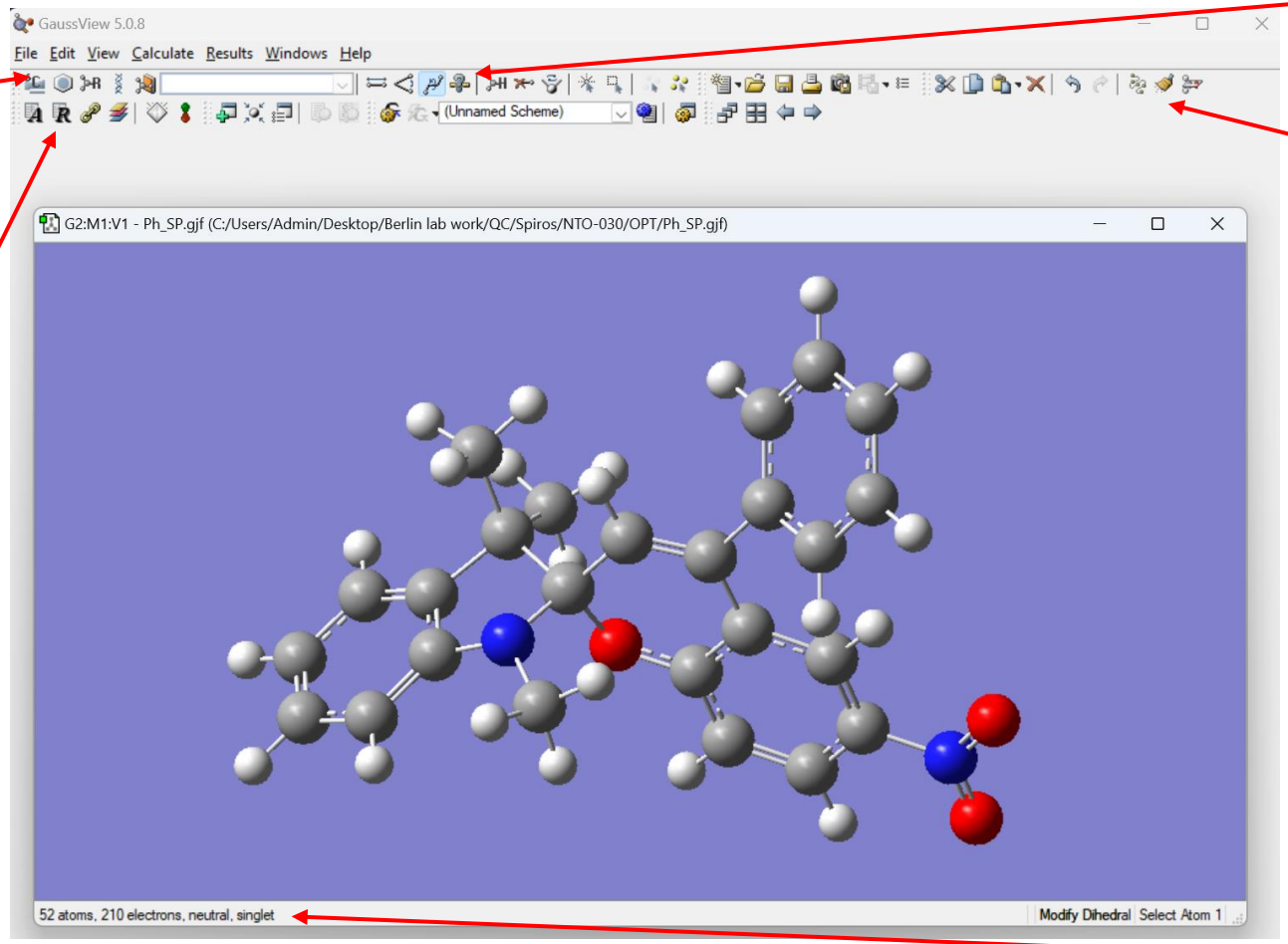
Selection of appropriately sized units can make up for easier and more intuitive understandement of scale and proportions. Kilograms are good for everyday life but widely impractical for particles, there mass is sometimes even expressed as energy ($E=mc^2$). (Mass of an electron = $9.1093837139\times10^{-31}$ kg, or 0.510 998 950 69 MeV => the well known value of 511 keV)

How to build an input and examples (GaussView)

Plenty of software can be used to generate the starting geometry and the basic shape of the input file.
However, everything can be done in text file and it is usually more convenient. (GaussView 5 as an example)

Selections of atoms or molecular fragments.

Editor of redundant coordinates. (Useful for scans and similar operations)

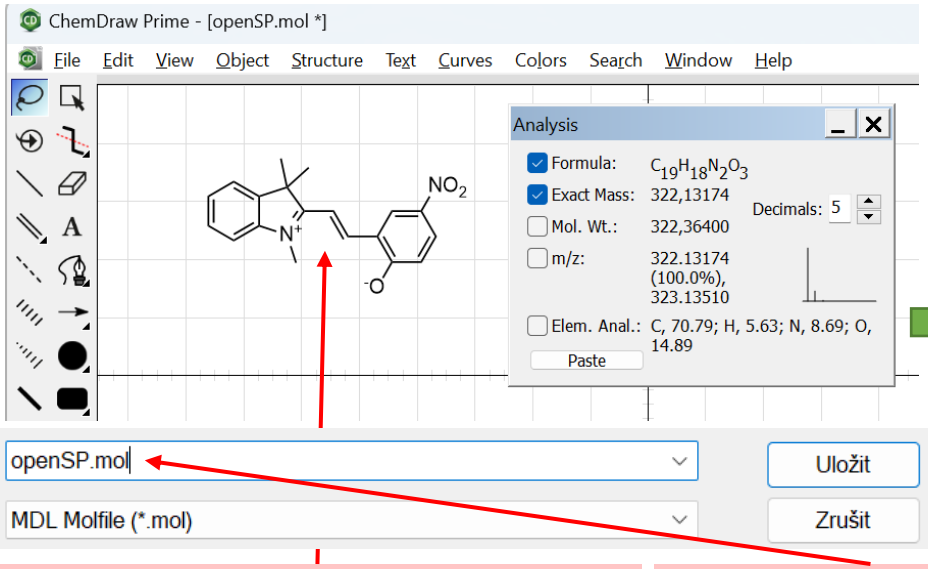


Tools to change bond length, angles, dihedral angles.

Crude preoptimization using molecular mechanics. (Can be detrimental for unusual structures)

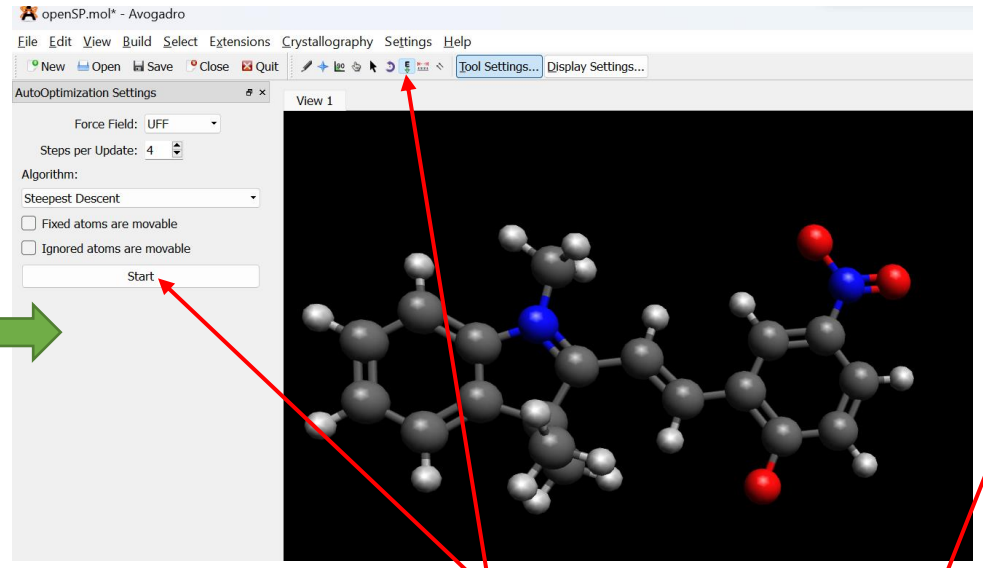
Electronic configuration, etc.

How to build an input using ChemDraw and Avogadro

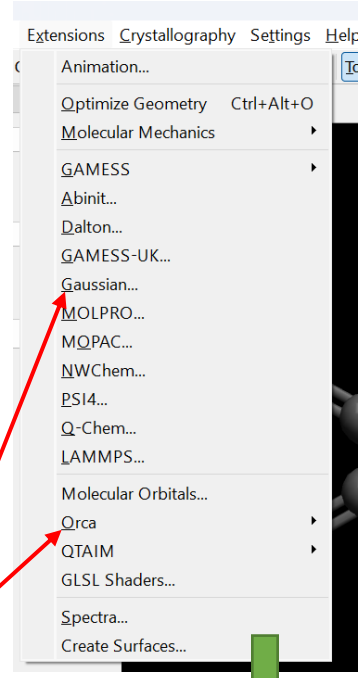


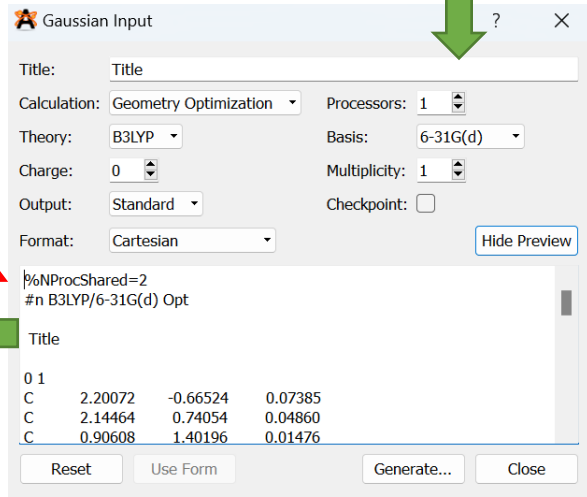
Draw your molecule in chemdraw.

Save it as .mol file.

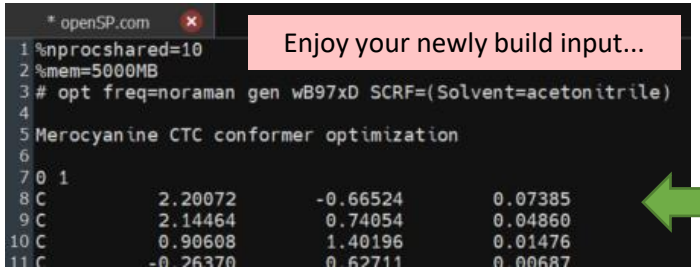


Open the file in Avogadro and use force field to preoptimize it! Afterwards, use the extension button and export the input file in the appropriate format (Gaussian, Orca, etc...)





Write the input for the calculations here or save the file and do it in the text file later!



Enjoy your newly build input...

Optimization (Gaussian)

Prior every investigation it is important to have the correct geometry of said molecule. It can be obtained via geometry optimization.

- Keywords # opt in Gaussian will do the trick.
- For well-behaved molecules, more or less every standard DFT functional will do a reasonable job
- Dispersion correction is essential! (# EmpiricalDispersion=GD3BJ for the latest variant in Gaussian, check the availability for functionals, or use dispersion corrected functionals. Orca contains the newest correction D4).
- Optimization can be directly coupled with frequency analysis to prove that minima was obtained. (Gaussian command # opt freq=noraman)

L3: opt freq=noraman is command for optimization of the molecule followed by frequency analysis. Noraman stands for not calculating Raman spectra (unless you want it).

```
1 %nprocshared=10
2 %mem=6000MB
3 # opt freq=noraman def2tzvp wB97xD SCRF=(Solvent=acetonitrile)
4
5 Description of the calculation goes here.
6
7 0 1
8  C                  4.72092500   -2.00832300   -0.83856400
9  C                  3.59228700   -1.49395800   -1.47348800
10 C                  2.85401600   -0.53425100   -0.80270500
11 C                  3.21818700   -0.10158200    0.47172700
12 C                  4.33243700   -0.61856900    1.09085300
13 C                  5.09136800   -1.58376700    0.42723800
14 H                  5.31185800   -2.76339200   -1.34434900
```

Optimization (Orca)

Prior every investigation it is important to have the correct geometry of said molecule. It can be obtained via geometry optimization.

- Keywords opt in Orca will do the trick.
- For well-behaved molecules, more or less every standard DFT functional will do a reasonable job
- Dispersion correction is essential! Orca contains plenty of functionals with D3BJ or D4 dispersion corrections. Besides that, functionals from Head-Gordon group (wB97 series) has often inbuilt dispersion corrections. Composite methods also contains dispesrion corrections by default.
- Optimization can be directly coupled with frequency analysis to prove that minima was obtained. (Keyword opt freq)

L1 contains the importat keywords to perform calculations. They can be further specify, if neccesary, bellow.

```
1   ! wb97x-D4 OPT Freq RIJCOSX def2-TZVP def2/J def2-TZVP/C TightSCF CPCM(Acetonitrile)
2   %maxcore 8000
3   %pal
4   nprocs 10
5   end
6   %scf
7       MaxIter 500
8   end
9
10  * xyz 0 1
11  6       -4.642144000     -2.085236000      0.322460000
12  6       -3.552236000     -1.706334000      1.105899000
13  6       -2.803250000     -0.619305000      0.689511000
14  6       -3.117416000      0.069411000     -0.480911000
```

Frequency analysis (Gaussian)

Frequency analysis in general serves 2 purposes. It tells information about at which point of potential energy surface a particular geometrical configuration of atoms lies (for example is it a minimum, transition state or just assortment of atoms which needs to be further optimized), and it is also used to get thermochemical data for particular system (with frequency analysis we can obtain zero-point energy (ZPE, enthalpy or Gibbs energy at particular temperature and pressure).

- Keyword in gaussian is # freq and can be either used alone or together with # opt command.
- It is relatively more computationally expensive compared to optimization.

L3: freq=noraman is command for standalone frequency analysis of particular system.

```
%nprocshared=10
%mem=3000MB
# freq=noraman gen scrf=(solvent=acetonitrile) wb97xD

Transition state of SP opening into CCC

0 1
6       -4.844819000     -1.777347000     -0.515129000
6       -4.370564000     -0.518015000     -0.903272000
6       -4.182348000     -2.521540000      0.462955000
6       -3.024912000     -2.036579000      1.080497000
```

Frequency analysis – Output (Gaussian)

Output concerning thermochemical data can be either find in the log file manually or through software (Chemcraft)
These values can be further used to assess relative stability of different species or to calculate reaction energies.
However, for this all calculations which are supposed to be compared needs to be done with the same method!

Output regarding thermochemistry found in a log file.

```
Zero-point correction=                           0.339080 (Hartree/Particle)
Thermal correction to Energy=                    0.358781
Thermal correction to Enthalpy=                  0.359725
Thermal correction to Gibbs Free Energy=         0.290484
Sum of electronic and zero-point Energies=          -1069.664844
Sum of electronic and thermal Energies=             -1069.645143
Sum of electronic and thermal Enthalpies=           -1069.644198
Sum of electronic and thermal Free Energies=        -1069.713440
```

Frequency analysis (Orca)

Frequency analysis in general serves 2 purposes. It tells information about at which point of potential energy surface a particular geometrical configuration of atoms lies (for example is it a minimum, transition state or just assortment of atoms which needs to be further optimized), and it is also used to get thermochemical data for particular system (with frequency analysis we can obtain zero-point energy (ZPE, enthalpy or Gibbs energy at particular temperature and pressure).

- Keyword in orca is # freq and can be either used alone or together with # opt command.
- It is relatively more computationally expensive compared to optimization.
- Some methods do not have analytical gradients implemented and thus numerical approach must be used. This is significantly more expensive.

L1: Freq is used to obtain Hessian.

Use numfreq command if analytical calculation is not available. Or if you need to obtain Raman spectra (In combination with %elprop Polar 1
end

```
! wb97x-D4 OPT Freq RIJCOSX def2-TZVP def2/J def2-TZVP/C TightSCF CPCM(Acetonitrile)
%maxcore 8000
%pal
nprocs 10
end

* xyz 0 1
6        4.201597000     -2.541695000     -0.449184000
```

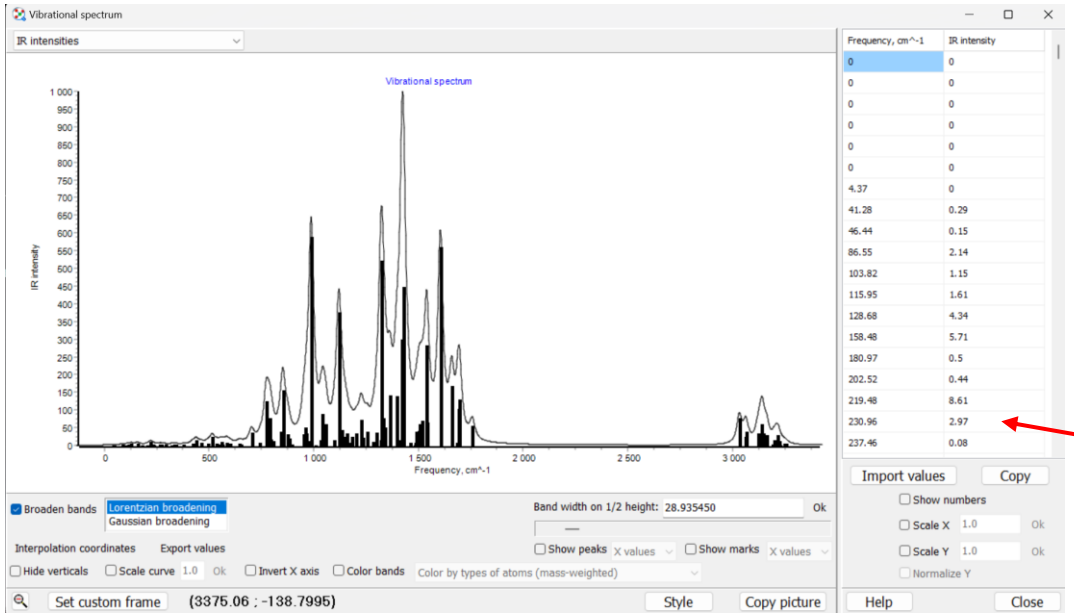
```
! wB97X-D3BJ OPT numfreq def2-TZVPP def2/J RIJCOSX Def2-TZVPP/C
%maxcore 11000
%pal
nprocs 20
end
%elprop Polar 1
end

* xyz 0 1
6       -0.487274000      0.224133000     -1.125710000
```

Frequency analysis – Output (Orca)

Output concerning thermochemical data can be either find in the log file manually or through software (Chemcraft)
These values can be further used to assess relative stability of different species or to calculate reaction energies.
However, for this all calculations which are supposed to be compared needs to be done with the same method!

Output regarding thermochemistry found in a out file. (Other thermochemical results are also available.)

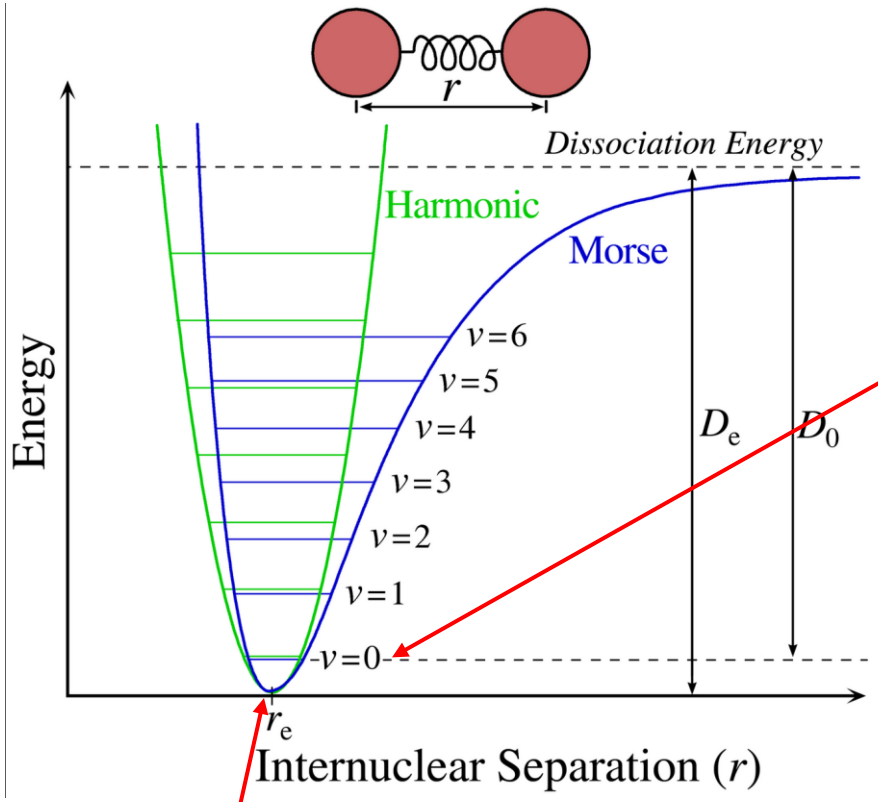
```
51756   ------------------
51757   GIBBS FREE ENERGY
51758   ------------------
51759
51760   The Gibbs free energy is G = H - T*S
51761
51762   Total enthalpy                  ...   -186.70164227 Eh
51763   Total entropy correction        ...     -0.06591574 Eh    -41.36 kcal/mol
51764   -----------------------------------------------------------------------
51765   Final Gibbs free energy         ...   -186.76755801 Eh
51766
51767   For completeness - the Gibbs free energy minus the electronic energy
51768   G-E(el)                         ...      0.29862944 Eh    187.39 kcal/mol
51769
```



IR or Raman can be simulated similarly to other spectral methods. Frequencies can be scaled by an appropriate value to give correct results (values typically around 0.96 for many methods), tables with scaling factor exists.

Difference in energies in the output

- Enthalpy and Gibbs free energy is pretty self-explanatory. But what does electronic energy (also known as SCF energy) and zero-point energy (ZPE) means?



Electronic energy

- Even upon reaching absolute zero, the system still has an energy associated with Heisenberg uncertainty principle: This lowest quantum oscillation is called ZPE.

Zero-point energy

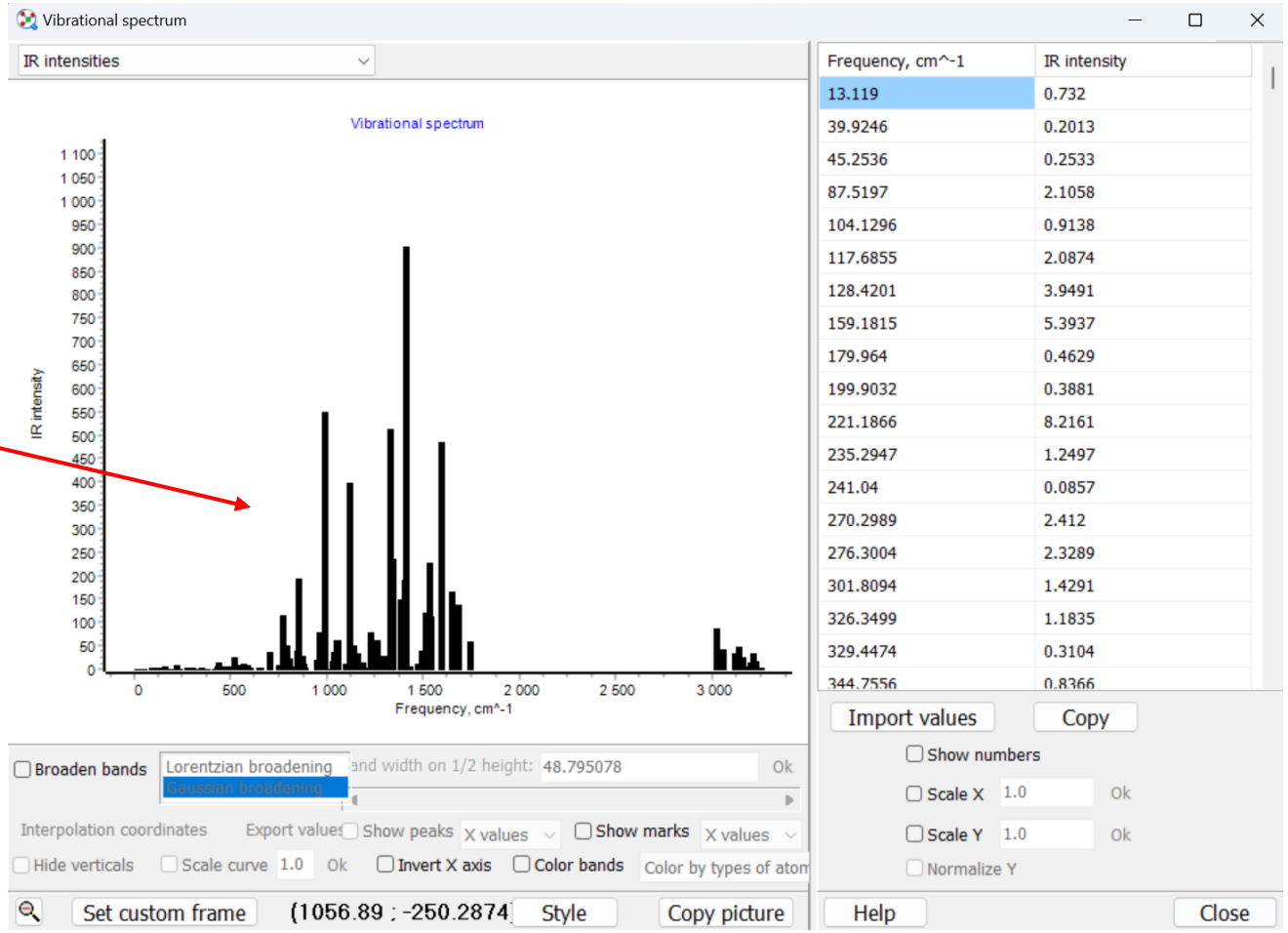
Differences in ZPE are the reason for kinetic isotope effect!

- The electronic energy represents the **total energy of the electrons in a given molecular system** in the context of the performer calculation.

Frequency analysis – Output

File should be also checked for the frequencies. If minima is required, no negative frequency should be present. For transition state, !1! negative frequency is expected.

Calculated IR spectra shown in Chemcraft. The list of frequencies on the right shows only positive values, therefore minima was obtained!



Local minimum vs global minimum

Does frequency analysis say something about if minimum is local or global?

- No! Mathematically they are both the same and don‘t contain imaginary frequencies. Global minimum is only the lowest one in energy.
- We can only compare minima of the same PES and say which one is lower or higher.

How to find global minimum?

- For a simple molecule, scan can help (see the example with butane in the scan section). This can tell us how energetics of the molecule are tied to certain geometrical parameter.
- There is no definitive proof that you have it, but for flexible molecules with plenty of conformers or rotamers, CREST can help.

Scans (Gaussian)

Scan is a useful tools for scanning a parameter of a molecule and see how it influences energy. Like bond length, angle or dihedral angle between planes.

L3: opt=modredundant will optimize the molecule withing the constraints of the desired scan parameters.

The scan of dihedral angle versus the energy of butane. In other words, how energetically demanding are different conformers?

D 11 8 5 1 73 5.00000
Meaning: D=scan dihedral angle
11 8 5 1 = numbers of atoms to define the dihedral angle
73 5.00000 = number of total steps and size of individual step

```
%nprocshared=4
%mem=4000MB
# opt=modredundant EmpiricalDispersion=GD3BJ TPSSTPSS/def2tzvp/def2tzv

Scan of bariers between conformers of butane

0 1
 C                 -2.80821973    0.36721380    0.00009151
 H                 -2.45291734   -0.64207277    0.00100229
 H                 -2.45019655    0.87058330   -0.87360097
 H                 -3.87821843    0.36865986   -0.00073612
 C                 -2.29487706    1.09327488    1.25743578
 H                 -2.65017945    2.10256145    1.25652501
 H                 -2.65290023    0.58990538    2.13112827
 C                 -0.75487893    1.09119363    1.25862695
 H                 -0.39685576    1.59456313    0.38493447
 H                 -0.39957655    0.08190706    1.25953773
 C                 -0.24153627    1.81725471    2.51597122
 H                 -0.59955944    1.31388521    3.38966370
 H                  0.82846243    1.81580865    2.51679885
 H                 -0.59683865    2.82654129    2.51506044

D 11 8 5 1 S 73 5.000000
```

Scan can be easily setup in GaussView via Redundant Coordinate Editor. But it can also easily be done by manually.
To setup different scans, you can use B for bond length scan and A for angle. Use 2 or 3 atom for definition respectively.

Scans (Orca)

Scan is a useful tools for scanning a parameter of a molecule and see how it influences energy. Like bond length, angle or dihedral angle between planes.

L3: opt=modredundant will optimize the molecule withing the constraints of the desired scan parameters.

The scan of dihedral angle versus the energy of butane. In other words, how energetically demanding are different conformers?

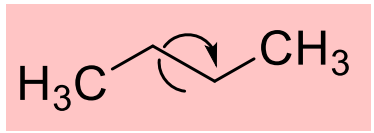
D 11 8 5 1 73 5.00000
Meaning: D=scan dihedral angle
11 8 5 1 = numbers of atoms to define the dihedral angle
73 5.00000 = number of total steps and size of individual step

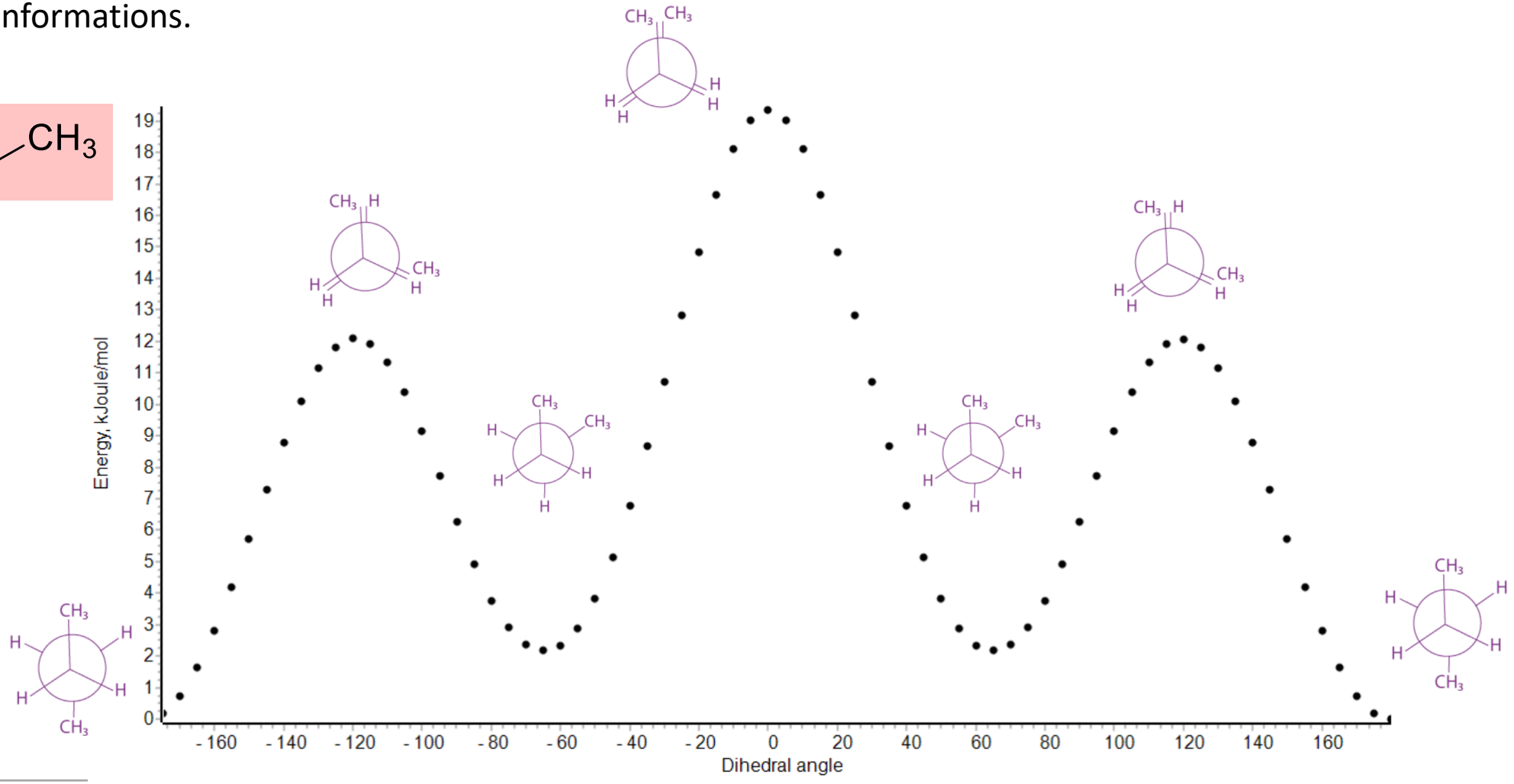
```
%nprocshared=4
%mem=4000MB
# opt=modredundant EmpiricalDispersion=GD3BJ TPSSTPSS/def2tzvp/def2tzv

Scan of bariers between conformers of butane

0 1
 C                 -2.80821973    0.36721380    0.00009151
 H                 -2.45291734   -0.64207277    0.00100229
 H                 -2.45019655    0.87058330   -0.87360097
 H                 -3.87821843    0.36865986   -0.00073612
 C                 -2.29487706    1.09327488    1.25743578
 H                 -2.65017945    2.10256145    1.25652501
 H                 -2.65290023    0.58990538    2.13112827
 C                 -0.75487893    1.09119363    1.25862695
 H                 -0.39685576    1.59456313    0.38493447
 H                 -0.39957655    0.08190706    1.25953773
 C                 -0.24153627    1.81725471    2.51597122
 H                 -0.59955944    1.31388521    3.38966370
 H                  0.82846243    1.81580865    2.51679885
 H                 -0.59683865    2.82654129    2.51506044

D 11 8 5 1 S 73 5.000000
```

Scan can be easily setup in GaussView via Redundant Coordinate Editor. But it can also easily be done by manually.
To setup different scans, you can use B for bond length scan and A for angle. Use 2 or 3 atom for definition respectively.

Scans - Output

Output in this case shows us the rotational barrier across the butane central C-C bond. The result is a famous diagrams showing maxima at eclipsed conformations and minima at gauche and antiperiplanar conformations.

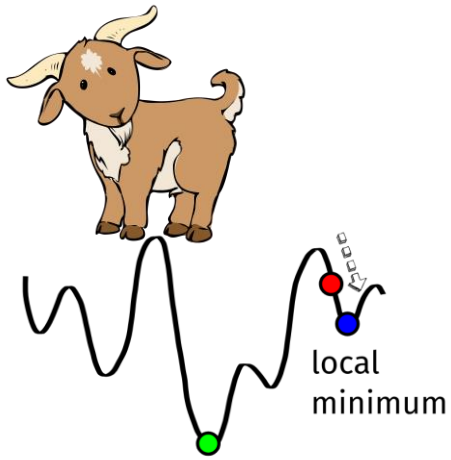




Conformer search in ORCA – GOAT: global geometry optimization and ensemble generator

ORCA version 6 and further contains a scheme to search for low lying conformers similar to the popular CREST system. It exploits a cheap xTB method and can use implicit solvent corrections. (**GOAT can easily hop over small hills**.)

- ! GOAT XTB alpb(acetonitrile) (*typical input to start a GOAT job*)



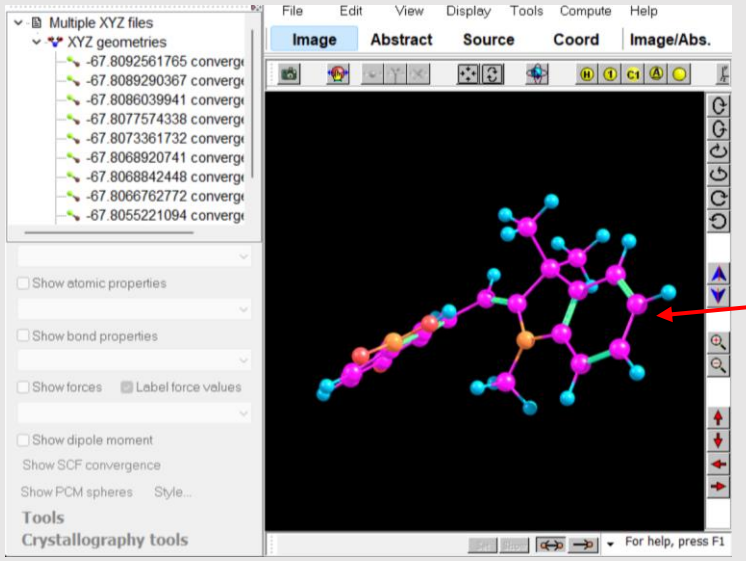
Keyword to start the conformer search. Solvent correction can be applied using efficient ALPB model. (Other solvent models are also available.)

```
! GOAT XTB alpb(acetonitrile)
%maxcore 8000
%pal
nprocs 10
end
%scf
    MaxIter 500
end

* xyz 0 1
   C         0.23180         0.91656        -0.10865
   C        -1.14231         0.23328         0.06003
```

Output

Found conformers, their coordinates and energies will be ordered by energy in a separate file „name“.finalensemble.xyz

TTT_GOAT.finalensemble.xyz



Open the file in Chemcraft to see all of them.

Transition states – TS berny

There are multiple ways how to obtain transition state. The most direct algorythm is TSBerny (Gaussian keyword # opt=(calcfc,ts). For this a geometry relatively close to the transition state is needed (can be obtained via scan for example). The algorithm then looks for a saddle point.

L3: opt=(calcfc,ts) is used as a straightforward way to obtain transition state.

```
%nprocshared=10
%mem=6000MB
# opt=(calcfc,ts) gen scrf=(solvent=acetonitrile) empiricaldispersion=gd3bj pbe1pbe

Spiropyran opening with DEF2-SVPD

0 1
 C                 -4.67629738   -0.88738301   -0.33501194
 C                 -4.08125924    0.32650330   -0.71024193
 C                 -3.94087651   -1.79363064    0.43041836
 C                 -2.62993475   -1.53162344    0.83177954
 C                 -2.07994553   -0.32637509    0.43937657
 C                 -2.78611896    0.59887735   -0.32335866
```

This method can be a bit tricky to apply since a good starting guess is neccessary. Alternative is QST2 (# opt=(calcfc,QST2)) where the desired transition state is being found on the way between starting material and product. Such input file requires 2 separet geometries!

Transition states – QST2

TSBerny method can be a bit tricky to apply since a good starting guess is neccessary. Alternative is QST2 (# opt=(calcfc,QST2)) where the desired transition state is being found on the way between starting material and product. Such input file requires 2 separate geometries!

L3: opt=(calcfc,QST2) is used to search for a saddle point between two minimum geometries.

The structure of starting material.

The structure of product.

```
%nprocshared=10
%mem=6000MB
# opt=(calcall,qst2) PBE1PBE EmpiricalDispersion=GD3BJ gen

Transtion state of spiropyran opening. SP to CCC

0 1
 C                 -4.56409100    2.15824300   -0.35439900
 C                 -3.45500300    1.77301300   -1.10463200
 C                 -2.74867800    0.65804300   -0.68798000
 C                 -3.12457800   -0.05312000    0.45044900
 C                 -4.21924500    0.33797500    1.18608100
 C                 -4.94607900    1.45731100    0.77810900
 H                 -5.13009700    3.03060900   -0.66049700
 H                 -3.15687600    2.33952500   -1.97823200
 N                  5.33723300    0.47830600    0.11973100
 O                  5.87035000    1.40462600    0.69561500
 O                  5.92599600   -0.35620400   -0.53761000
 H                  1.66443300   -2.66103400   -1.45683800

CCC

0 1
 C                 -4.25702900    2.50811400   -0.45496000
 C                 -2.99419700    2.11070800   -0.88033900
 C                 -2.60691800    0.82154500   -0.57360500
 C                 -3.41559300   -0.05709900    0.13336400
 C                 -4.66745100    0.34595000    0.54472400
 C                 -5.08462400    1.64055400    0.24517100
 H                 -4.59531600    3.51458600   -0.66980300
 H                 -2.34477200    2.79514300   -1.41079500
 H                 -5.31607300   -0.32432500    1.09766900
```

Transition states – NEB (Orca)

Nudged elastic band method in combination with transition state optimization is a fast algorithm to obtain transition state geometry between two minima.
The used input is reminiscent of QST2 method from Gaussian, but internal working principles differ. To use this method one need geometry of the starting material and the product (in a separate file).

L1: NEB-TS calls for this type of transition search. Can be combined with FREQ keyword to immediately obtain frequency calculation to confirm the saddle point.

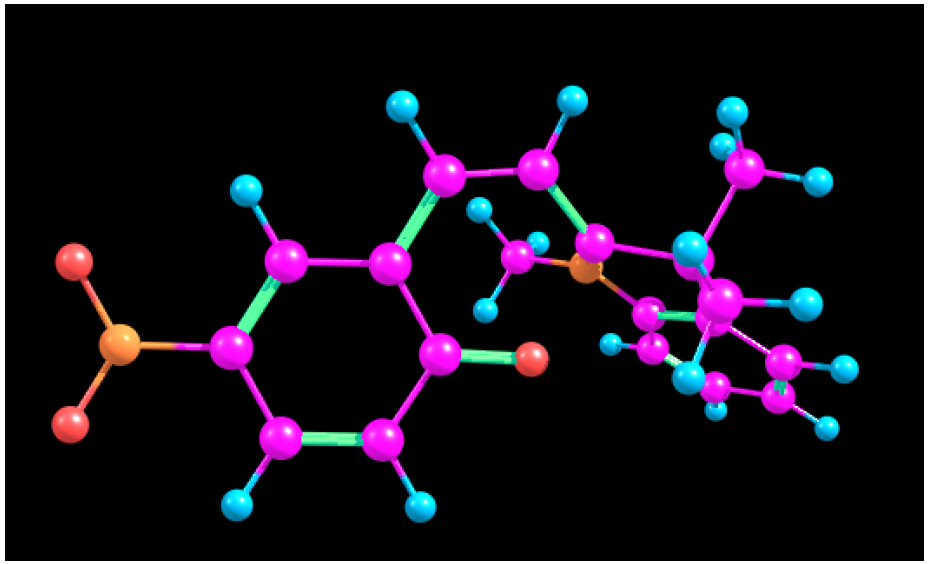
L7: the file name containing the geometry of product. In needs to be in the same folder as the input file.

```
1  ! r2scan-3c NEB-TS CPCM(Acetonitrile)
2  %maxcore 8000
3  %pal
4  nprocs 10
5  end
6  %neb
7  neb_end_xyzfile"CCC.xyz"
8  end
9
10 * xyz 0 1
11   C   -4.64359297415015      -2.09661542489847      0.33215181750188
12   C   -3.54427607696554      -1.71643224041853      1.11055659843610
```

Transition states – NEB output (Orca)

The output will contain the geometry of the transition state and frequency analysis (if requested!).

To confirm the saddle point (transition state), exactly 1 imaginary frequency must be present! The value should be relatively high on the order of several hundreds cm^{-1}. Otherwise, the transition state might be only something like a methyl group rotation.



To further confirm that the transition state belongs to the claimed transformation, intrinsic reaction coordinate (IRC) calculation should be used!

Transition states – IRC pathway (Orca)

Intrinsic reaction coordinate is a type of calculation where a molecular geometry in transition state is let to relax downhill to its starting and product forms. This is useful to confirm that between the investigated starting material and product is only one saddle point.

The keyword is „irc“ and can be followed by several other to improve the optimization algorithms (KDIIS, …) and FREQ to once again confirm the presence of only one imaginary frequency in the TS and to give imformation in which direction is the downhill path.

L1: IRC is used to call for the intrinsic reaction coordinate search.

```
1   !wb97x-3C Freq IRC cpcm(acetone)
2   %maxcore 8000
3   %pal
4   nprocs 10
5   end
6   %scf
7       MaxIter 500
8   end
9
10  * xyz 1 1
11  6       -8.845836000      3.809593000      0.145167000
12  53      -6.346699000      4.207625000     -0.598934000
```

Transition states – Intrinsic reaction coordinate (IRC) (Gaussian)

In order to prove that a particular transition state is directly connected with the starting molecule and the product of interest and the path doesn‘t contain any other intermediates, IRC can be performed. This is done by taking the geometry of the desired TS and let the program follow the path of least resistance down and see, whether the outcome of this calculation leads to expected structures. If it doesn‘t, there is likely another intermediate and TS in the way or the TS structure is wrong.

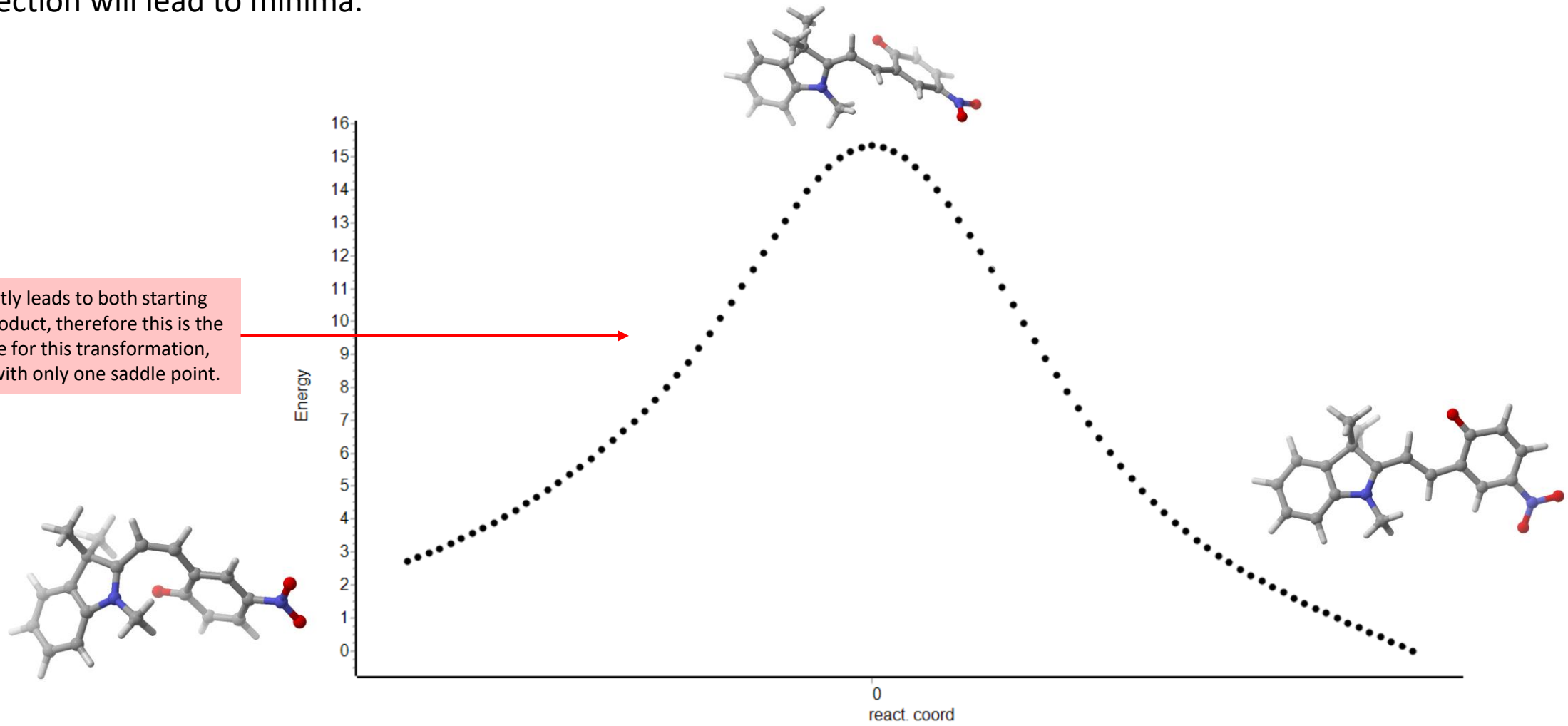
L3: irc=(maxpoints=50,calcfc) is used in order to get the coordinate. If the path is longer, more points than 50 might be neccessary. Calfc is used in order to calculate the initial direction of the descent from the saddle point.

```
%nprocshared=10
%mem=5000MB
# irc=(maxpoints=50,calcfc) freq=noraman tpsstpss/def2tzvp/def2tzv EmpiricalDispersion=GD3BJ
scrf=(solvent=acetone)

IRC for Finkelstein reaction of methylene chloride with iodide

-1 1
6       -1.174762000      0.000220000      0.000018000
1       -1.247062000      1.028990000     -0.307105000
1       -1.247717000     -0.780149000     -0.737272000
1       -1.247386000     -0.248130000      1.044508000
17      -3.729590000     -0.000054000     -0.000005000
53       1.399882000     -0.000021000     -0.000003000
```

Transition states – Intrinsic reaction coordinate (IRC) – Output (Gaussian)

The output can be again visualized in Chemcraft and should contain the transition state as the highest point and both direction will lead to minima.

The pathway directly leads to both starting structure and the product, therefore this is the real transition state for this transformation, and it is a reaction with only one saddle point.

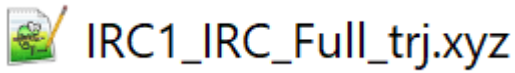


Transition states – IRC pathway (Orca)

The output of IRC calculation is a list of of coordinates and their energies on the downhill path. The list of coordinates and energies can be found in the output file.

The trajectory allone can be found in the out. file.

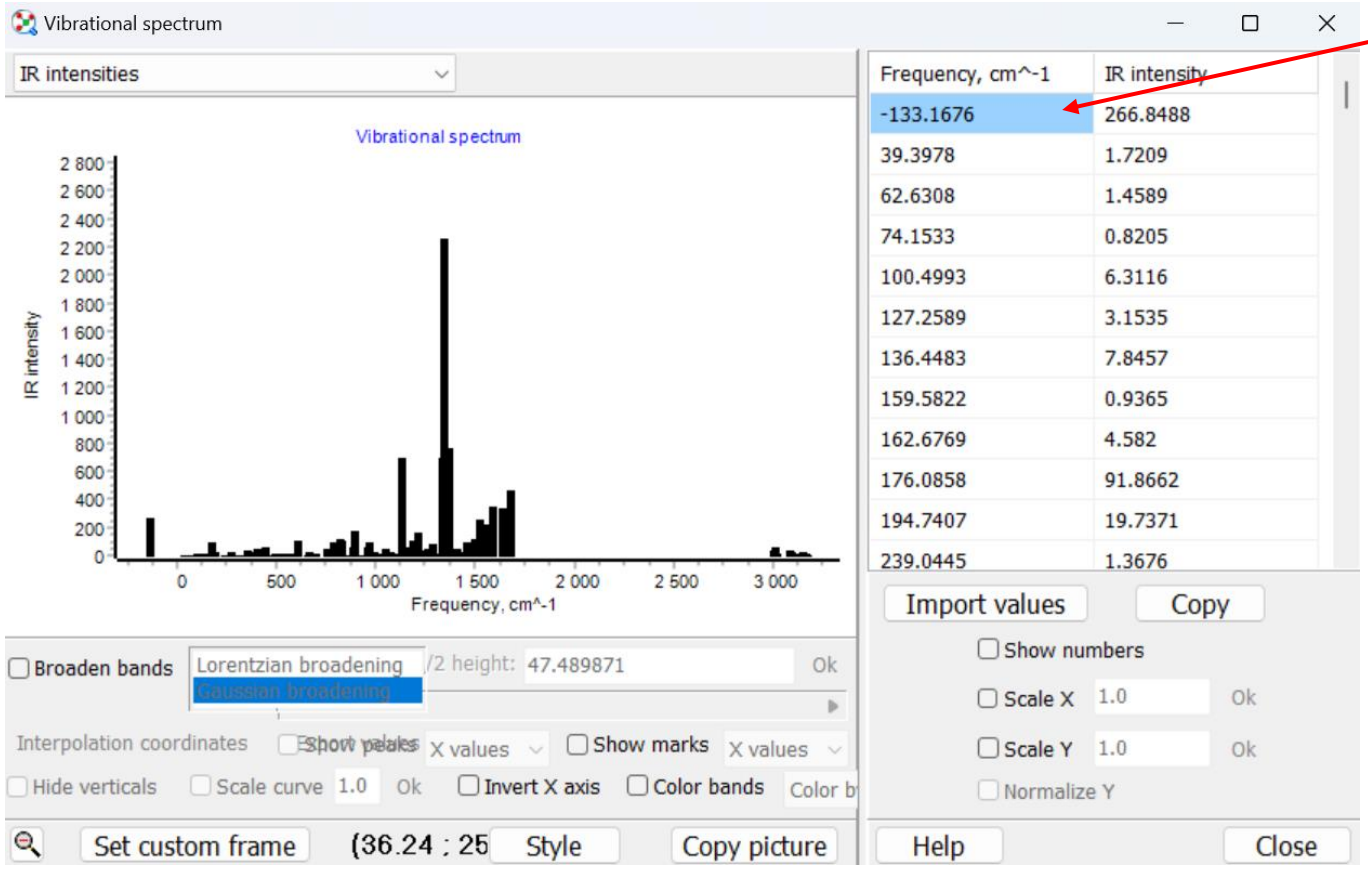
The file containing all the relevant geomtries is named [input_name]_IRC_Full_trj.xyz




```
                        ************************************************************
                        *                        FORWARD IRC                       *
                        ************************************************************


Iteration    E(Eh)      dE(kcal/mol)  max(|G|)   RMS(G)
Convergence thresholds                0.002000  0.000500
   0       -33.520921   18.004451    0.035392  0.013419
   1       -33.529361   12.708242    0.049224  0.016285
   2       -33.540455    5.746531    0.053460  0.017986
   3       -33.551870   -1.416671    0.052502  0.018249
   4       -33.563568   -8.756861    0.054612  0.018475
   5       -33.575139  -16.018064    0.054280  0.017921
   6       -33.586014  -22.842489    0.049723  0.016077
   7       -33.595074  -28.527210    0.038144  0.012034
   8       -33.600613  -32.003343    0.016025  0.005412
   9       -33.602294  -33.057802    0.009641  0.003063
  10       -33.603623  -33.892028    0.005883  0.002163
  11       -33.604364  -34.356680    0.006666  0.002719
  12       -33.604509  -34.447874    0.003559  0.001855
  13       -33.604728  -34.585478    0.004078  0.001784
  14       -33.605015  -34.765465    0.004794  0.001989
  15       -33.605370  -34.988470    0.005621  0.002453
  16       -33.605758  -35.231635    0.006412  0.002639
  17       -33.606222  -35.522600    0.007292  0.003140
  18       -33.606713  -35.831041    0.008108  0.003294
  19       -33.607284  -36.189335    0.008995  0.003808

                        ************************************************************
                        *  MAXIMUM NUMBER OF ITERATIONS REACHED - STOPPING IRC RUN  *
                        ************************************************************



                        ************************************************************
                        *                        BACKWARD IRC                      *
                        ************************************************************


Iteration    E(Eh)      dE(kcal/mol)  max(|G|)   RMS(G)
Convergence thresholds                0.002000  0.000500
   0       -33.581741  -20.160747    0.047075  0.016581
   1       -33.590603  -25.721702    0.032219  0.011255
   2       -33.595331  -28.688564    0.011231  0.004374
   3       -33.596916  -29.683335    0.011592  0.003810
   4       -33.598702  -30.804204    0.012288  0.003575
   5       -33.600692  -32.052637    0.011768  0.003431
   6       -33.602430  -33.143158    0.009594  0.002819
   7       -33.603640  -33.902781    0.005871  0.002133
   8       -33.603834  -34.024453    0.004942  0.001724
   9       -33.603957  -34.101338    0.004189  0.001366
  10       -33.604122  -34.205239    0.002955  0.001064
  11       -33.604284  -34.306486    0.002220  0.001046
  12       -33.604465  -34.420428    0.002684  0.001288
  13       -33.604669  -34.548541    0.003200  0.001430
  14       -33.604920  -34.706167    0.003753  0.001720
  15       -33.605196  -34.879007    0.004305  0.001864
  16       -33.605515  -35.078921    0.004764  0.002124
  17       -33.605855  -35.292490    0.005211  0.002252
  18       -33.606233  -35.529848    0.005583  0.002483
  19       -33.606631  -35.779324    0.005942  0.002598

                        ************************************************************
                        *  MAXIMUM NUMBER OF ITERATIONS REACHED - STOPPING IRC RUN  *
                        ************************************************************
```

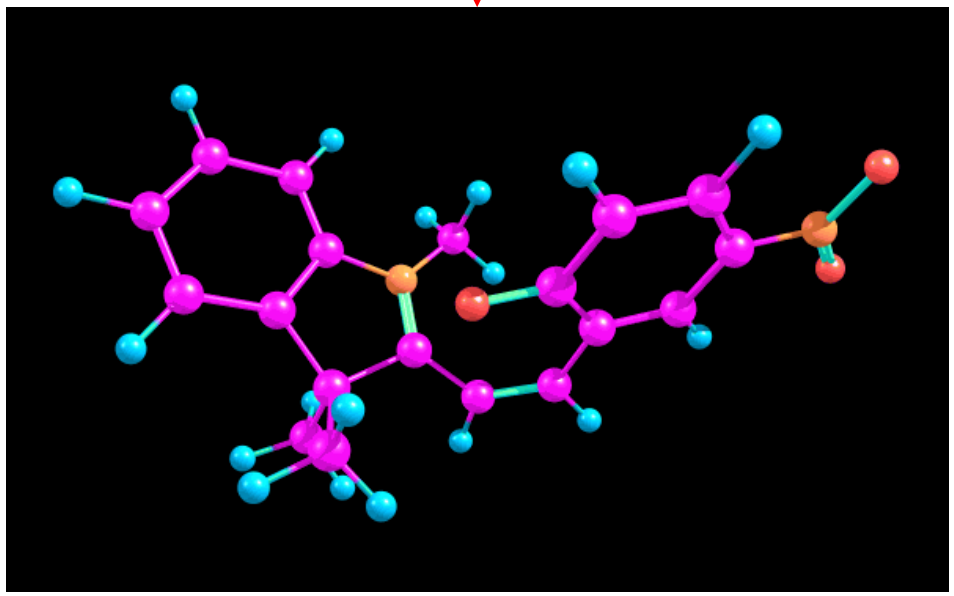
Transition states – frequency analysis

Just like with minima, transtion state needs to be confirmed via frequency analysis. This is done the exact same way – only in this case the outcome should contain 1 negative frequency in the output.



Negative frequency corresponding to the TS as shown in chemcraft!

Frequencies can be animated to show how does the path look. Bellow shown transition state of spiropyran opening.



Vertical excitation energies (TD-DFT)

Time dependent density functional theory is a way to calculate vertical excitation energies. It is simpler and faster then multireference methods, however, it has limitations of accuracy for certain systems. Specifically, molecules with high charge transfer character are typically not calculated accurately. On the other hand, triplet energies of aromatic compounds can be estimated rather well.

Use TD-DFT with caution and ideally compare the obtained data to experimental results. Alternatively use post Hartree-Fock method to calculate UV-Vis spectra.

Vertical excitation energies (TD-DFT) - Gaussian

You can calculate absorption spectra or triplet energies in a sense of vertical excitation (Frank-Condon principle) using time-dependent density functional theory (TD-DFT) formalism.

L3: td=(50-50,nstates=20) This command will perform calculation of vertical excitation for the molecular geometry from the input. You can specify to calculate singlets or triplets only (keyword singlets or triplets respectively) or mixture of both by using command 50-50 as shown in this example. The command nstates=20 decides how many vertical state excitations will be searched for.

L7: 0 1 means like in the case of any other calculation, what is the state of origin. Thus in this case, ground state closed shell singlet is considered. But you can easily change this for ionic molecule, radicals or triplet state (or higher multiplicities).

```
1 %nprocshared=10
2 %mem=4000MB
3 # td=(50-50,nstates=20) pbe1pbe def2tzvp
4
5 Azulene absorption spectra
6
7 0 1
8  C                 -1.96967661   -1.46052089    0.57859573
9  C                 -0.60894661   -1.46052089    0.57859573
10 C                  0.27977839   -0.35843389    0.57859573
11 C                 -2.87447561   -0.37160089    0.57940073
12 C                 -0.02755761    0.96283911    0.57886573
13 C                 -2.61364661    0.96304611    0.58028873
14 C                 -1.36146361    1.61385011    0.57995573
15 H                 -2.46035461   -2.45613589    0.57801673
16 H                 -0.11093061   -2.45066189    0.57825773
17 H                  1.35703839   -0.62382189    0.57818273
18 H                 -3.94282061   -0.66731389    0.57950173
19 H                 -3.48658461    1.64820911    0.58116973
20 C                  0.94393539    2.04940711    0.57852773
21 H                  2.02077139    1.89617111    0.57753273
22 C                  0.26396939    3.25107611    0.57951573
23 H                  0.70081039    4.24887211    0.57957173
24 C                 -1.15694161    2.99058611    0.58040873
25 H                 -1.92598961    3.75983711    0.58134073
26
```

Vertical excitation energies (TD-DFT) – Gaussian - Results

The resulting log file contains the information about the excited states and contains the multiplicity of the excited state, energy of said state and in case of singlet state, a parameter called force constant, which is directly proportional to extinction coefficient.

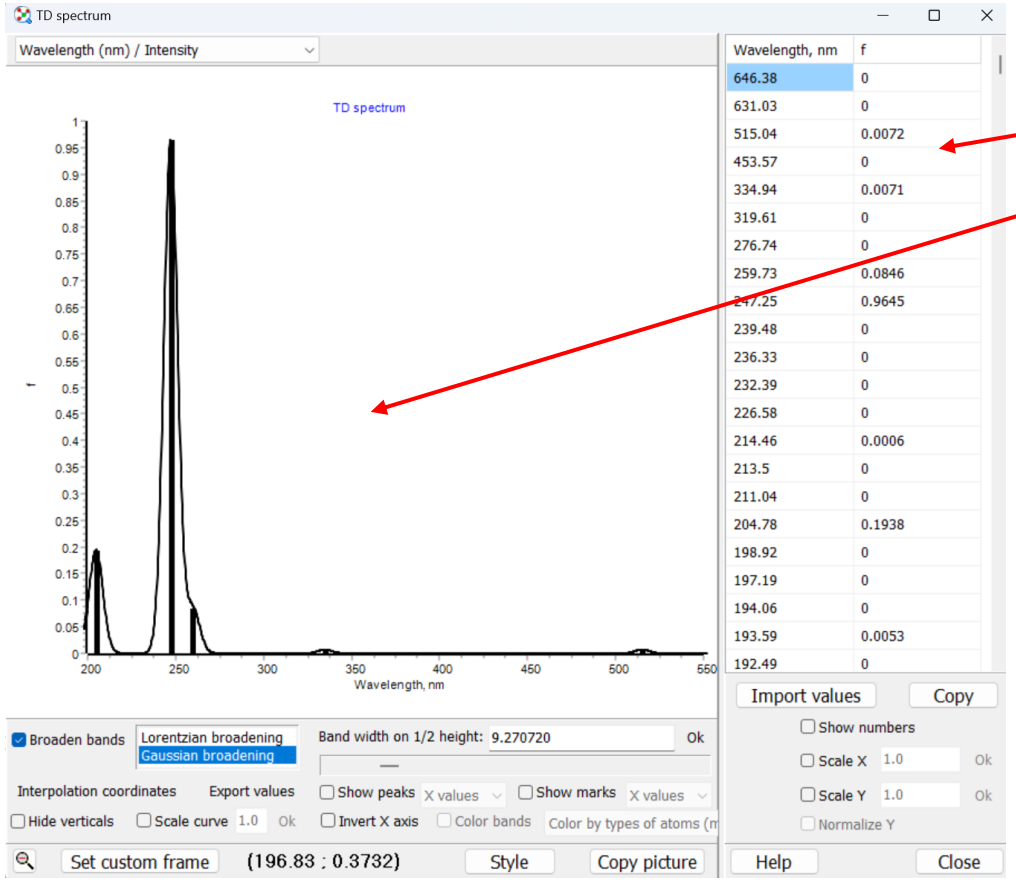
Besides the information about the nature of the state, energy, etc... You can also see the compostion of the orbitals involved in each transition and thus say something about the nature of this excitation.

We can see from these singlet excitation states that S1 has a very low oscilator strength (force constant) and thus its absorbtion should be very low. On the other hand, the Excited state 8 corresponds to an order of magnitude larger oscillator strength.

```
 Excitation energies and oscillator strengths:

 Excited State   1:      Triplet-A      1.9181 eV  646.38 nm  f=0.0000  <S**2>=2.000
      34 -> 35         0.70169
 This state for optimization and/or second-order correction.
 Total Energy, E(TD-HF/TD-DFT) =  -385.427825103
 Copying the excited state density for this state as the 1-particle RhoCI density.

 Excited State   2:      Triplet-A      1.9648 eV  631.03 nm  f=0.0000  <S**2>=2.000
      33 -> 35         0.42348
      34 -> 36         0.58026
      33 <- 35         0.12674
      34 <- 36         0.15178

 Excited State   3:      Singlet-A      2.4073 eV  515.04 nm  f=0.0072  <S**2>=0.000
      34 -> 35         0.70070

 Excited State   4:      Triplet-A      2.7335 eV  453.57 nm  f=0.0000  <S**2>=2.000
      33 -> 35         0.56146
      34 -> 36        -0.41256

 Excited State   5:      Singlet-A      3.7017 eV  334.94 nm  f=0.0071  <S**2>=0.000
      33 -> 35         0.46336
      34 -> 36         0.53249

 Excited State   6:      Triplet-A      3.8793 eV  319.61 nm  f=0.0000  <S**2>=2.000
      32 -> 35        -0.13793
      33 -> 36         0.68397

 Excited State   7:      Triplet-A      4.4802 eV  276.74 nm  f=0.0000  <S**2>=2.000
      32 -> 35         0.67708
      33 -> 36         0.14193

 Excited State   8:      Singlet-A      4.7736 eV  259.73 nm  f=0.0846  <S**2>=0.000
      32 -> 35         0.25694
      33 -> 35        -0.13127
      33 -> 36         0.62361
      34 -> 36         0.11401

 Excited State   9:      Singlet-A      5.0146 eV  247.25 nm  f=0.9645  <S**2>=0.000
      33 -> 35         0.51326
      33 -> 36         0.18203
      34 -> 36        -0.44810

 Excited State  10:      Triplet-A      5.1771 eV  239.48 nm  f=0.0000  <S**2>=2.000
      30 -> 35        -0.36906
      30 -> 36         0.16826
      33 -> 39        -0.19487
      34 -> 39         0.51926
```

Vertical excitation energies (TD-DFT) – Gaussian - Results

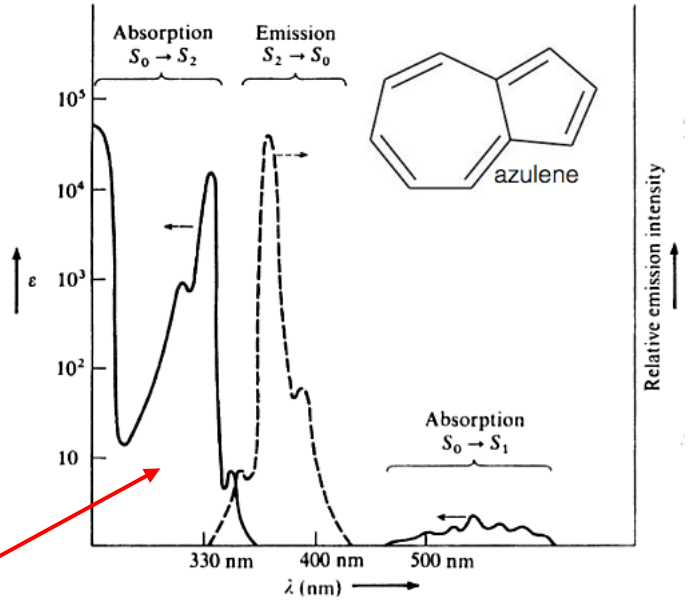
We can also visualize the data through multiple programs. Here, Chemcraft is used. The results is not in a form of classical spectra with bands but rather just lines corresponding to each transition. Artificial Gaussian or Lorentzian broadening can be applied later.



List of transitions.

Modeled spectra with artificial broadening.

The experimentally measured spectra.



TDDFT - Orca

- simulate UV/vis spectra on DFT level
- for vertical single electronic excitation

input

- Basis set
- Functional
- Solvent
- Number of excitations (roots)

```
================================================================================
                                       INPUT FILE
================================================================================
NAME = orca.inp
|  1> ! ma-def2-TZVPP
|  2> ! BHANDHLYP
|  3> ! CPCM(Acetonitrile)
|  4> %tddft
|  5> nroots 15
|  6> end
|  7> %pal
|  8> nprocs 8
|  9> end
| 10> %maxcore 2000
| 11> *xyzfile 0 1 orcaopt.xyz
| 12>
| 13>
| 14>                          ****END OF INPUT****
================================================================================
```

Number of processors

Memory for each process, here: 2000 MB

your .xyz file

charge

multiplicity

* Example for calculation with orca. orca does NOT! Include triplet states as a default. Please consult manual for that. Tamm-Dancoff approximation is used by default, can be switched off.

TDDFT - Orca

- simulate UV/vis spectra on DFT level
- for vertical single electronic excitation

output

- Number of excitations (roots/states)
- Excitation energy
- Probability of excitation (f_{osc}) = oscillator strength

ABSORPTION SPECTRUM VIA TRANSITION ELECTRIC DIPOLE MOMENTS

State	Energy (cm-1)	Wavelength (nm)	fosc	T2 (au**2)	TX (au)	TY (au)	TZ (au)
1	23160.3	431.8	0.138465440	1.96821	1.24903	0.58250	-0.26236
2	28243.6	354.1	0.034362832	0.40054	0.60063	-0.01006	0.19920
3	30973.5	322.9	0.733789596	7.79932	-2.78716	-0.04601	-0.17020
4	33716.6	296.6	0.027184831	0.26544	0.39826	0.14247	-0.29415
5	34888.0	286.6	0.004597881	0.04339	0.13012	0.15389	-0.05267
6	35721.8	279.9	0.005201679	0.04794	0.05058	-0.20067	0.07150
7	36757.3	272.1	0.079779634	0.71454	0.78627	0.22545	-0.21329
8	38391.9	260.5	0.006544208	0.05612	0.13116	0.01203	-0.19690
9	41231.3	242.5	0.055729282	0.44497	-0.65023	-0.14759	-0.01970
10	41914.0	238.6	0.013380326	0.10510	0.30794	0.05405	-0.08571
11	42113.9	237.5	0.031018237	0.24248	-0.48728	0.00474	0.07076
12	43446.8	230.2	0.394743932	2.99112	-1.71763	-0.09531	-0.17829
13	44282.9	225.8	0.053212676	0.39560	-0.54128	-0.04491	0.31717
14	44654.4	223.9	0.160343957	1.18213	-0.79777	-0.54134	0.50263
15	45758.6	218.5	0.036180634	0.26030	0.26540	0.31841	-0.29745

* Example for calculation with orca

TDDFT – Orca

- Open orca.out file with Chemcraft

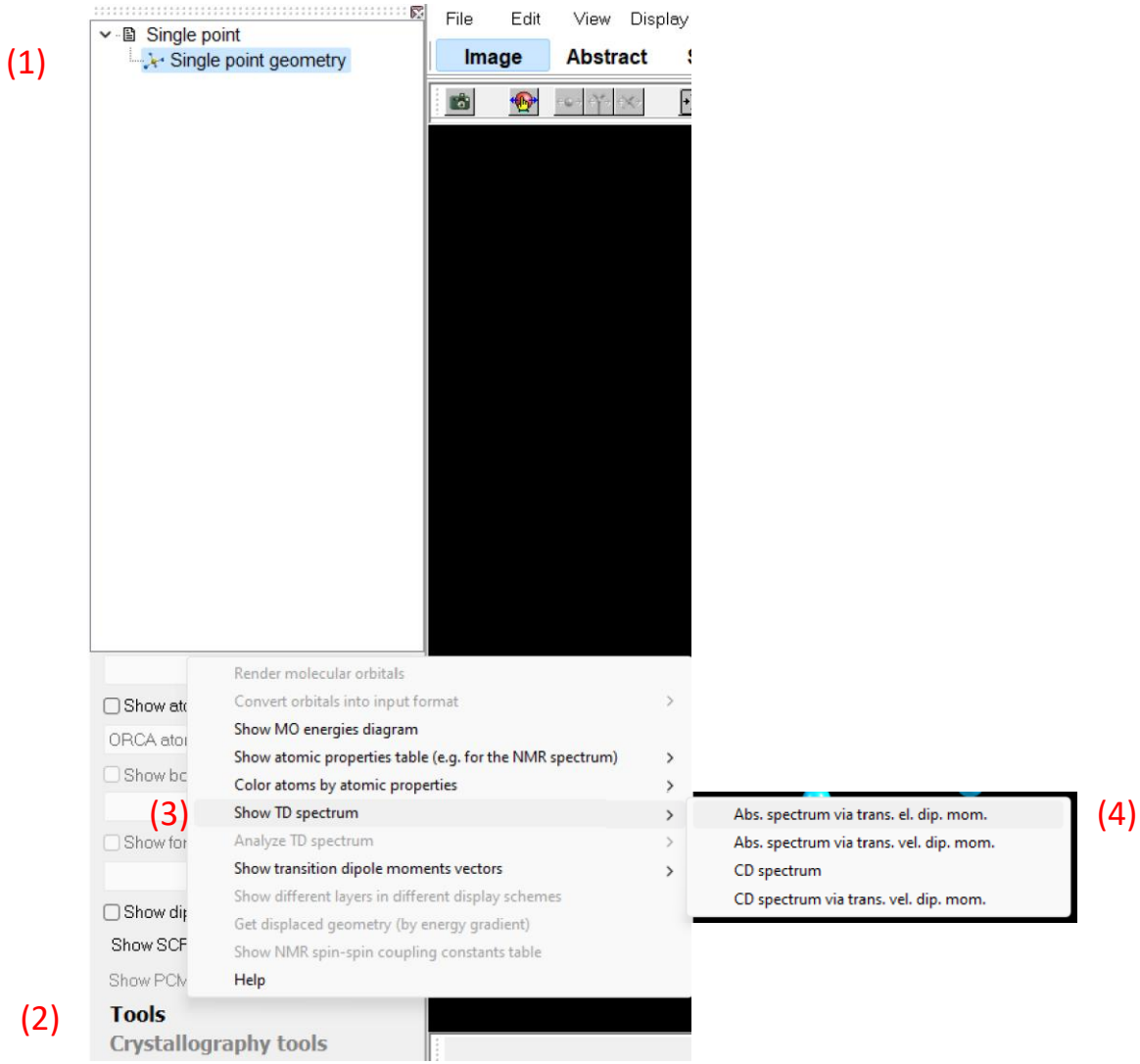
(1) select "single point geometry"

(2) Tools

(3) Show TD spectrum

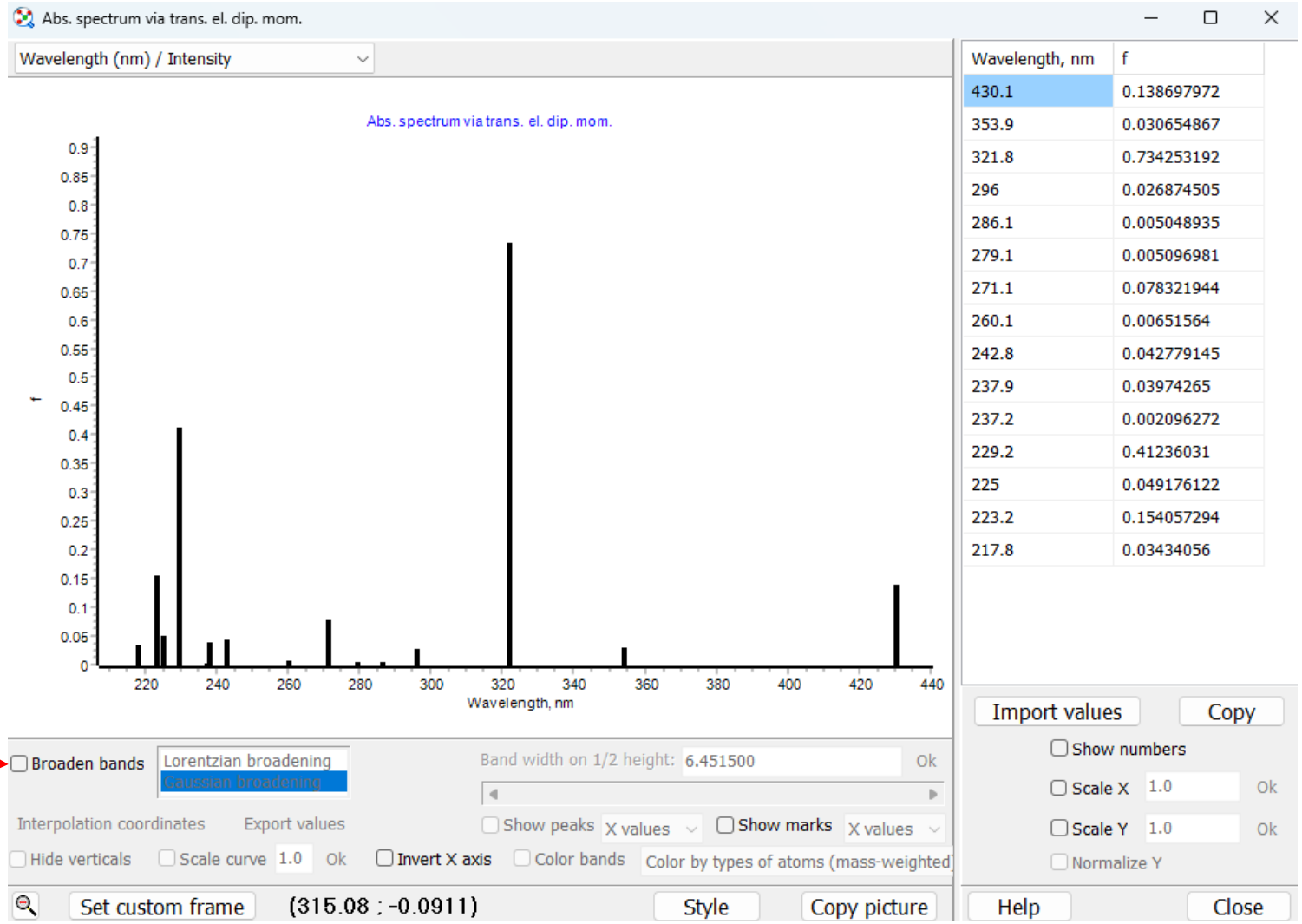
(4) Abs. Spectrum via trans. el. dip. mom.

- New window will open (see next slide)



* Example for calculation with orca

TDDFT – Orca



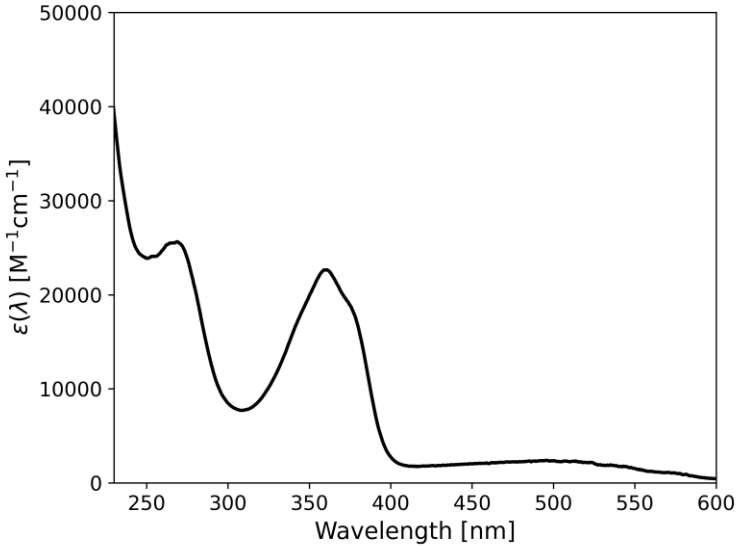
Wavelength, nm	f
430.1	0.138697972
353.9	0.030654867
321.8	0.734253192
296	0.026874505
286.1	0.005048935
279.1	0.005096981
271.1	0.078321944
260.1	0.00651564
242.8	0.042779145
237.9	0.03974265
237.2	0.002096272
229.2	0.41236031
225	0.049176122
223.2	0.154057294
217.8	0.03434056

Broadening the lines to a peak shape with empirical Gaussian or Lorentzian expansion function

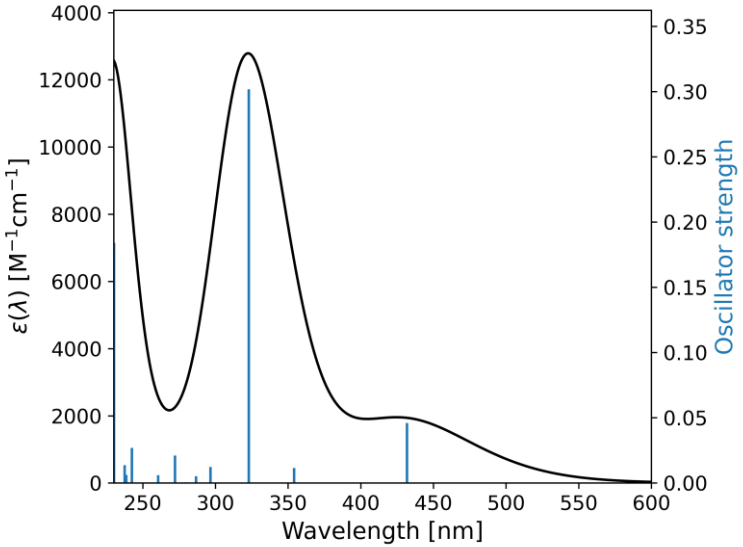
* Example for calculation with orca

TDDFT - Orca

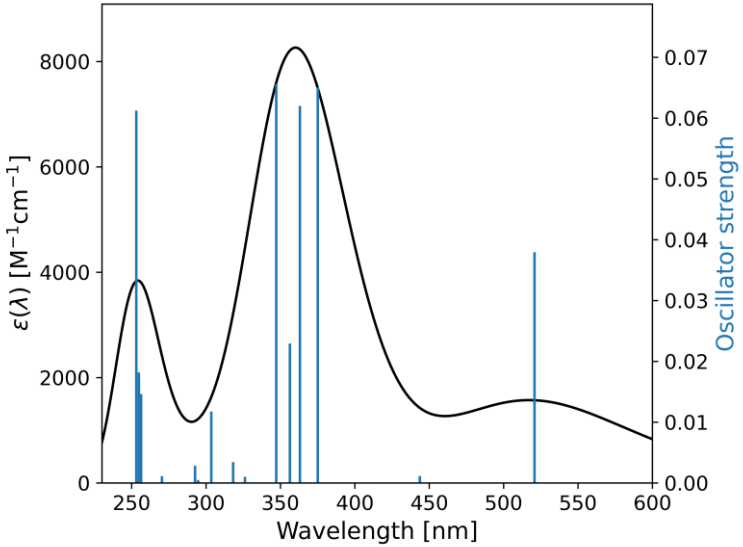
- Carefull with functionals!



Measured UV/Vis



BHHLYP



PBE0

- Use experimental data to find the right functional !
- Remember: you are only calculating electronic states, not vibrational (measured spectra is combination)

* Example for calculation with orca

Molecular properties

In addition of transition states, energies, scans, UV-vis spectra simulation, one can also calculate other properties :

- HOMO-LUMO gaps
- MO electronic densities
- Alpha spin densities
- Atomic charges (Mulliken, Natural charges)
- Bond orders
- Orbital occupancies

Accessible through population analysis

- Polar surfaces

Accessible through Cube function in Gaussview

Molecular properties : Rendering MOs→Population analysis

Canonical MOs represent the electronic structure of the entire system, including all atoms and bonds.
→ get MO energies, HOMO-LUMO gap…(**Pop=Full)**

Localized MOs are a transformation of canonical MOs creating orbitals localized around specific regions of the molecule. (**Pop=NBO**)
→ used for chemical interpretation (they provide a clearer picture of bonding interactions within the molecule), finding bond order, atomic charges…

HOW TO guide :

Step 1 : Optimise geometry of molecule at desired level of theory ex: PBE0/def2TZVPD

Step 2 : Take the output geometry as input for the population analysis, specify the same level of theory and add the keyword : Pop=Full (canonical), Pop=NBO (Natural Bond Orbitals)

Step 3 : Find other desired information in .log file (Atomic Charges, Bond Orders, Orbital Occupancies, HOMO-LUMO gap, geometry SP energy) or by using other analysis algorithms.

Step 4 : Render desired MO isosurface electronic density (have .fchk file)

Molecular properties : Step 2 : example of input file : formic acid

For visualization of the MOs, save the .chk files because the visual information is stored there (not in .log file)
Input file example : The geometry of the input file should be previously optimized (STEP 1) at the same level of theory (here : PBE0-D3BJ/def2TZVP) !

```
%chk=FA_NBO.chk
%NProcShared=8
%mem=3000MB
#P PBE1PBE def2tzvp pop=(full,nboread,savenbo) gfoldprint

FA_NBO

0 1
C          0.00000        0.38115        0.00000
O          1.17271        0.19456        0.00000
O         -0.89158       -0.62094        0.00000
H         -0.46332        1.38590        0.00000
H         -1.78573       -0.26178        0.00000

$NBO BNDIDX $END
```

Input for NBO visualization

```
%chk=FA_Full.chk
%NProcShared=8
%mem=3000MB
#P PBE1PBE def2tzvp Pop=Full

FA_Full

0 1
C          0.00000        0.38115        0.00000
O          1.17271        0.19456        0.00000
O         -0.89158       -0.62094        0.00000
H         -0.46332        1.38590        0.00000
H         -1.78573       -0.26178        0.00000
```

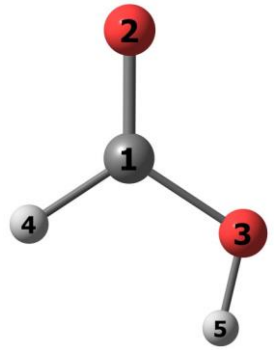
Input for canonical MO visualization

%chk=FA_Full.chk → Saves the .chk file (ACHTUNG ! Same name of the .chk file as for the .log file!)
Pop=Full → indicates to the program to perform a canonical population analysis
pop=(full,nboread,savenbo) gfoldprint and **$NBO BNDIDX $END** at the end of the coordinates allows NBO visualization

Output opened in ChemCraft:
FA_Full.log : SP energy, HOMO-LUMO gap, energy diagram of the different MOs and other info
FA_Full.chk :WF/density info that allows to draw the isosurface of the MO of choice (HOMO,LUMO,HOMO-1 for example).

Molecular properties : Step 3 : NBO.log file information !

Example : Formic Acid



Atomic Charges: charge distribution among the atoms in your molecule. Positive charges indicate electron deficiency, while negative charges indicate electron excess. (Mulliken charges **A** / Natural charges **B**)

Orbital Occupancies: how many electrons occupy each MO. (Natural Population : Core, Valence, Rydberg distribution of electrons **C**)

Bond Orders: bond orders between pairs of atoms, indicating the strength of the chemical bonds (1=single bond, 2= double bond…). Different manners of calculating them : molecular orbital (MO) theory (**D**), natural atomic orbital (NAO) theory (**E**) or Wiberg Bond Index (**F**).

Mulliken charges:

	Atom	1
1	C	0.186373
2	O	−0.267088
3	O	−0.317460
4	H	0.075056
5	H	0.323120

A

Sum of Mulliken charges = −0.00000

Mulliken charges with hydrogens summed into heavy atoms:

	Atom	1
1	C	0.261429
2	O	−0.267088
3	O	0.005659

Summary of Natural Population Analysis:

B **C**

Atom	No	Natural Charge	Natural Population Core	Valence	Rydberg	Total
C	1	0.61489	1.99964	3.34466	0.04081	5.38511
O	2	−0.52720	1.99973	6.49655	0.03091	8.52720
O	3	−0.65037	1.99976	6.62249	0.02812	8.65037
H	4	0.08429	0.00000	0.91165	0.00405	0.91571
H	5	0.47839	0.00000	0.51922	0.00240	0.52161
* Total *		0.00000	5.99914	17.89458	0.10628	24.00000

Natural Population

Core	5.99914 (99.9856% of	6)
Valence	17.89458 (99.4143% of	18)
Natural Minimal Basis	23.89372 (99.5571% of	24)
Natural Rydberg Basis	0.10628 (0.4429% of	24)

Atom	No	Natural Electron Configuration
C	1	[core]2S(0.88)2p(2.46)3S(0.01)4p(0.02)
O	2	[core]2S(1.70)2p(4.80)3d(0.02)
O	3	[core]2S(1.69)2p(4.94)3d(0.02)
H	4	1S(0.91)
H	5	1S(0.52)

C

MO bond order:

D

	Atom	1	2	3	4	5
1.	C	0.0000	1.7895	−0.3280	0.8803	−0.0721
2.	O	1.7895	0.0000	0.1720	−0.2484	−0.1754
3.	O	−0.3280	0.1720	0.0000	−0.2078	−0.0868
4.	H	0.8803	−0.2484	−0.2078	0.0000	−0.0095
5.	H	−0.0721	−0.1754	−0.0868	−0.0095	0.0000

Atom-atom overlap-weighted NAO bond order:

E

	Atom	1	2	3	4	5
1.	C	0.0000	1.4008	0.9283	0.7857	0.0221
2.	O	1.4008	0.0000	−0.0523	−0.0279	0.0039
3.	O	0.9283	−0.0523	0.0000	−0.0410	0.6379
4.	H	0.7857	−0.0279	−0.0410	0.0000	0.0029
5.	H	0.0221	0.0039	0.6379	0.0029	0.0000

Wiberg bond index matrix in the NAO basis:

F

	Atom	1	2	3	4	5
1.	C	0.0000	1.8444	1.0590	0.9010	0.0026
2.	O	1.8444	0.0000	0.1696	0.0705	0.0158
3.	O	1.0590	0.1696	0.0000	0.0271	0.7549
4.	H	0.9010	0.0705	0.0271	0.0000	0.0002
5.	H	0.0026	0.0158	0.7549	0.0002	0.0000

The canonical population analysis doesn't contain bond order information (only Mulliken charges) !

gaussian specific example

Molecular properties : Step 4 : Render MOs

1. In order to render the canonical MOs or NBOs, one has to convert the .chk file into a .fchk file (readable by post-treatment softwares like Chemcraft).

To do this, run in the cluster terminal the following commands as follow :

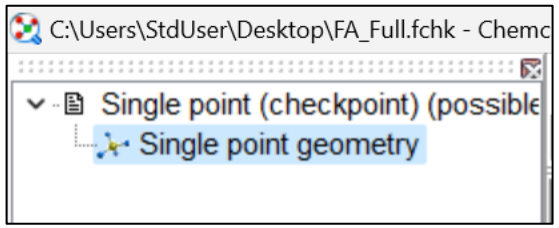
```
turcasiu@chccs:~> module load G16-A03
turcasiu@chccs:~> formchk filename.chk filename.fchk
```

A converted **.fchk** file appears in one's directory.

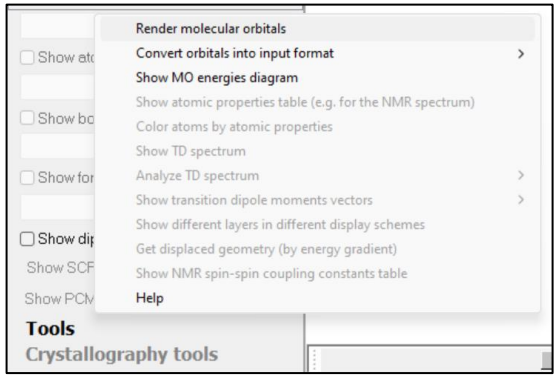
Molecular properties : Step 4 : Render MOs

2. Open the .fchk file with Chemcraft.

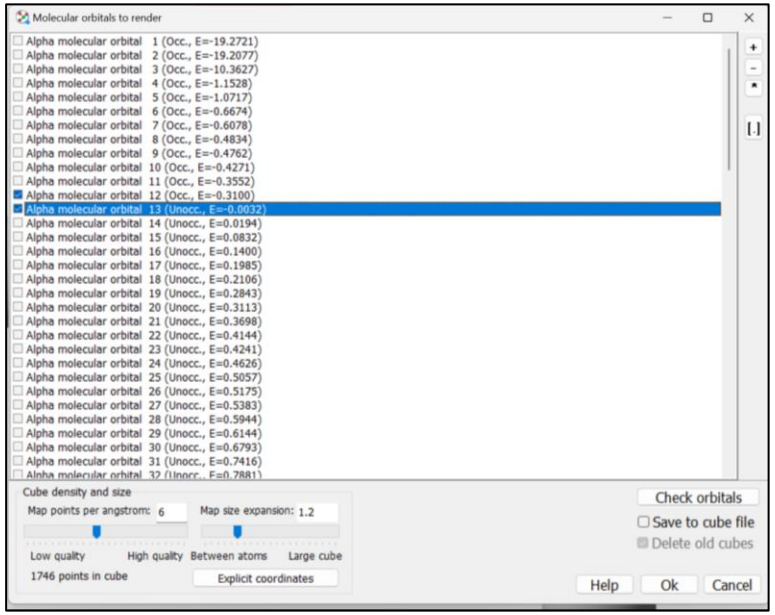
3. Click on the optimised geometry



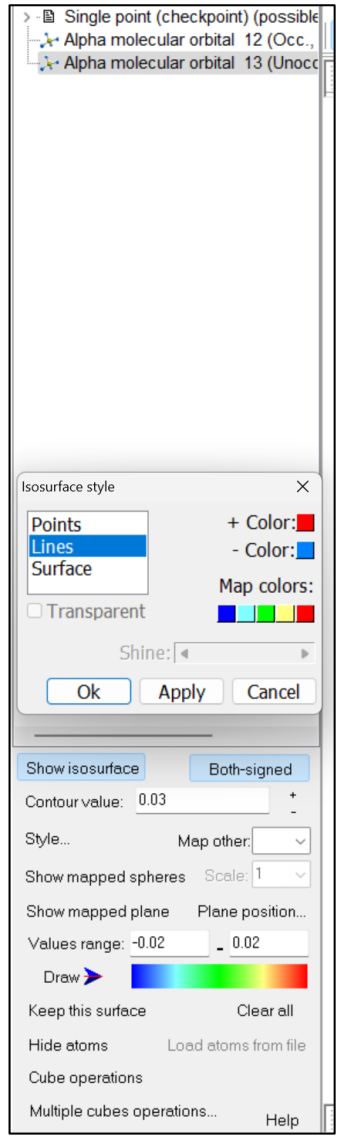
4. Tools → Render Molecular orbitals



5. Choose to image the MO of choice between all the computed ones (here HOMO and LUMO are chosen).

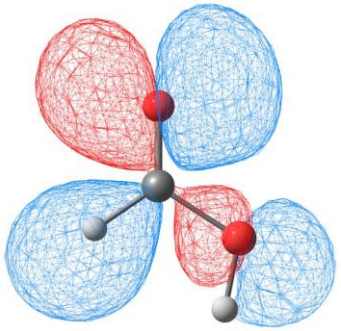


6. Modify the visualization parameters of the surface (show both signs, design…etc)

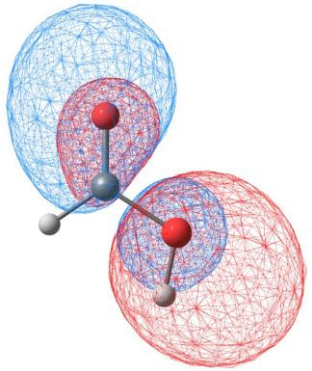


Molecular properties : Ex : canonical vs NBO of formic acid

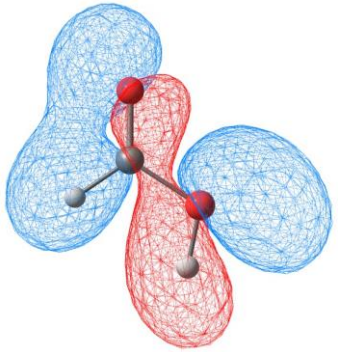
HOMO



HOMO-1



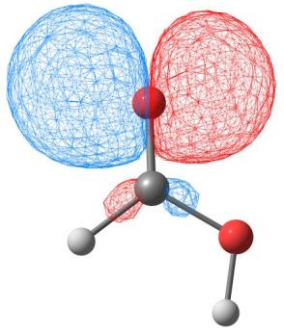
HOMO-2



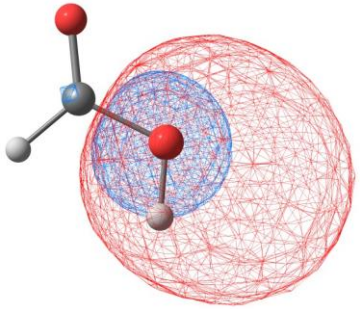
Canonical MOs are delocalized (appear as a mixture of nearly all AO of the molecule) which makes them harder to interpret in terms of chemical bonds or reactivity

←Canonical vs NBO →

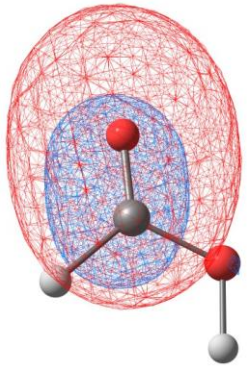
NBO's are more localized and follow our "chemical intuition" in terms of **π** and ***σ***-bonds, lone pairs and localization of electrons on atoms.



HOMO



HOMO-1



HOMO-2

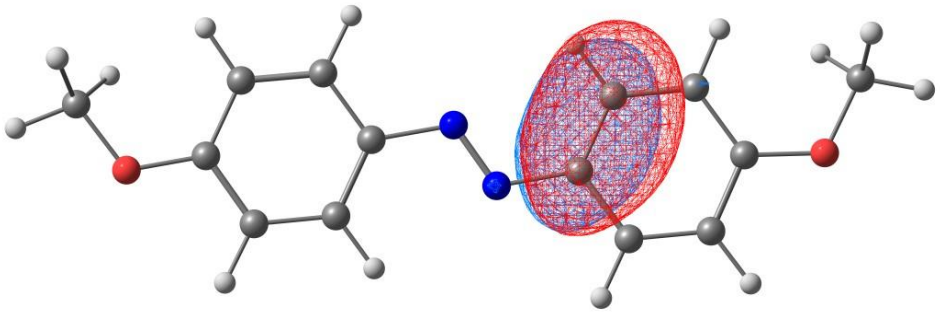
Molecular properties : Alpha spin densities

The electronic density of the alpha electron is that of the **SOMO (single occupied molecular orbital) :**

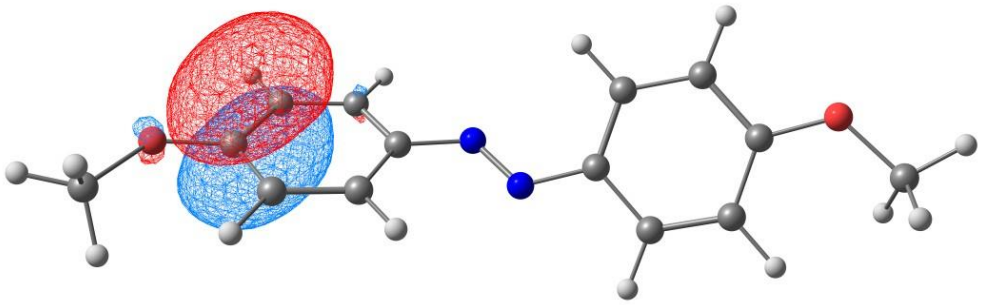
- **The procedure is the same as before (regular MOs)**
- **Reminder :** use unrestircted DFT and do not forget to ajust the multiplicity in the input (2 for a radical)

```
%chk=A-rE_NBO.chk
%NProcShared=8
%mem=3000MB
#P uPBE1PBE EmpiricalDispersion=GD3BJ def2tzvp pop=(full,nboread,savenbo) gfoldprint

A-rE_NBO

1 2
N          -0.43484        0.67937       -0.55719
```

Example of input for rendering NBOs of a radical cation



E azo HOMO (NBO)



E $^{\cdot +}$ azo α-SOMO (NBO)

gaussian specific example

Solvent models : The problem of not having isolated molecules

Unisolated molecules → Interactions with their environment !

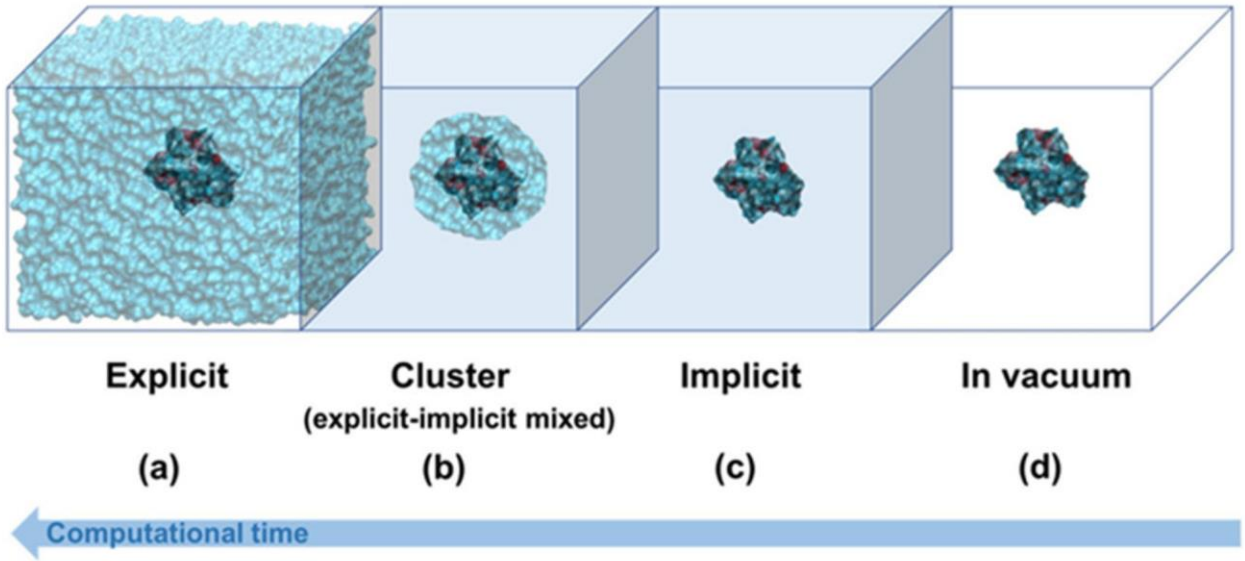
$$E_{interact}=E_{intra}+E_{inter}$$

$$E_{intra}=E_{strech} + E_{bend} + E_{torsion} + E_{cross} + E_{out}$$ → Bonded interactions

$$E_{inter}=E_{elect} + E_{vdW}$$ → Non-bonded interactions (interaction of electronic densities)

Non-bonded interactions can be of long-range or short-range type.

Implicit and **explicit** solvation models exist.



Solvent models : Implicit/Continuum solvation models

Polarizable Continuum Model (PCM) :

→ Uses : **molecular cavity** following the real geometry of the investigated system + surface charge distribution to represent the polarization of the environment

→ Easy to use and a computationally cheap methodology

→ by extending the solute so to contain few explicit solvent molecules, PCM can be used when short-range solute–solvent interactions are important

Solvation model density (SMD) :

→ based on **empirical paramerization** by Truhlar group (more detailed model accounting for a broader range of solvation effects). Specifically optimized for the Minnesota functionals

→ can be more computationnaly intensive

One way how to select a solvatation model is to take a look at what model was used to parametrize the particular functional by their authors.

Remarks :

→ PCM and SMD : most widely used and versatile models, good accuracy across a range of solvation problems.

→ Need to benchmark different solvation models against experimental data to assess their performance for a particular application.

→ Other solvation models exist like Conductor-like screening model (COSMO/COSMO-RS)

Review on Polarizable continuum model (**PCM**) : WIREs Comput Mol Sci 2012, 2: 386-404 https://doi.org/10.1002/wcms.1086
SMD solvation model : *J. Phys. Chem. B* 2009, 113, 18, 6378–6396 https://doi.org/10.1021/jp810292n
Review on **COSMO** and COSMO-RS solvation models : WIREs Comput Mol Sci 2018, 8:e1338. https://doi.org/10.1002/wcms.1338

Solvent models : Implicit/Continuum solvation models

Input for Polarizable Continuum Model (PCM) :

```
%NProcShared=8
%mem=6000MB
# opt freq=noraman PBE1PBE EmpiricalDispersion=GD3BJ def2tzvp SCRF=(PCM,solvent=acetonitrile)

A-E_ACN

0 1
N          -0.37446         0.65977        -0.00015
N           0.34550        -0.35658        -0.00011
```

Input for Solvation model density (SMD) :

```
%NProcShared=8
%mem=6000MB
# opt freq=noraman PBE1PBE EmpiricalDispersion=GD3BJ def2tzvp SCRF=(SMD,solvent=acetonitrile)

A-E_ACN

0 1
N          -0.37446         0.65977        -0.00015
N           0.34550        -0.35658        -0.00011
```

The solvent contribution to the energy (ΔG_{solv}) is the difference between the energy in vacuum and the energy in solution is given by :

$$\Delta G_{solv} = E_{vacuum} - E_{solvent}$$

List of available solvents on : https://gaussian.com/scrf/ click “More” then “Solvents”

gaussian specific example

Solvent models : Explicit solvation models

Explicit models are more complicated, more computationally demanding but necessary if one needs a precise and accurate description of hydrogen-bonding systems or short-range solvent-molecule interactions.

Difficulty in computing explicit solvation shells because of the large number of molecules required.

Bare-minimum solvation model : Put solvent molecules up to saturation of the coordination sites of the system in addition of using a PCM/SMD model

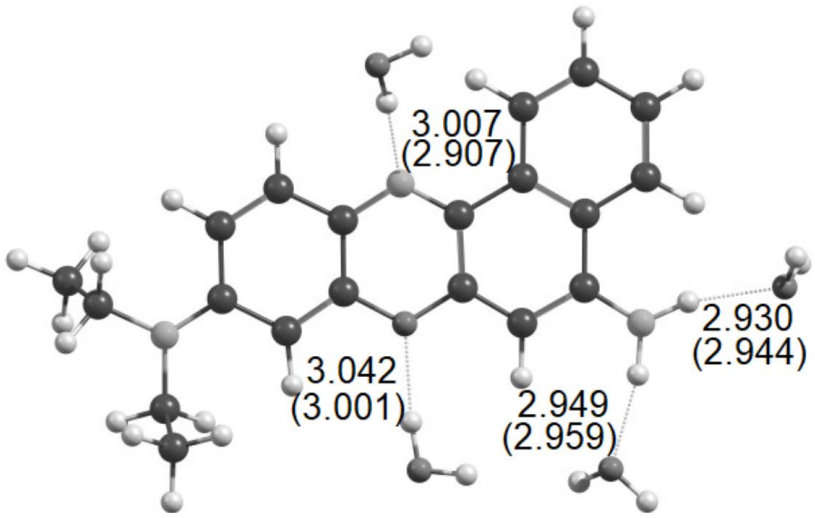
- Use CREST's crystal growth functionality and specify the numbers of solvent molecules desired to form the solvation shell
- Then opimization of energy with Gaussian or ORCA

(Alternatively, when the system is simple one can also just manually put several solvent molecule at the expected coordination sites)

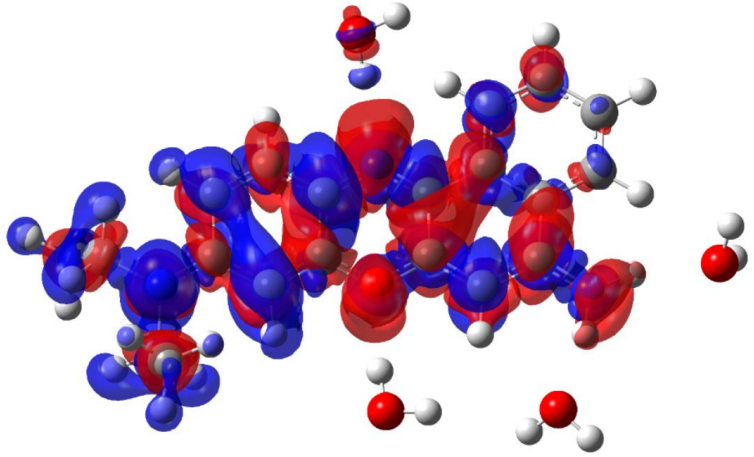
Full explicit model : Crystal growth

Solvent models : Bare minimum solvation examples

Example of Nile Blue :
Most probable cationic charge site based on the closeness of the coordinating 4 water molecules to it



Electron density difference between ground state and excited state : One water molecule seems to be more involved than the others in the transition !

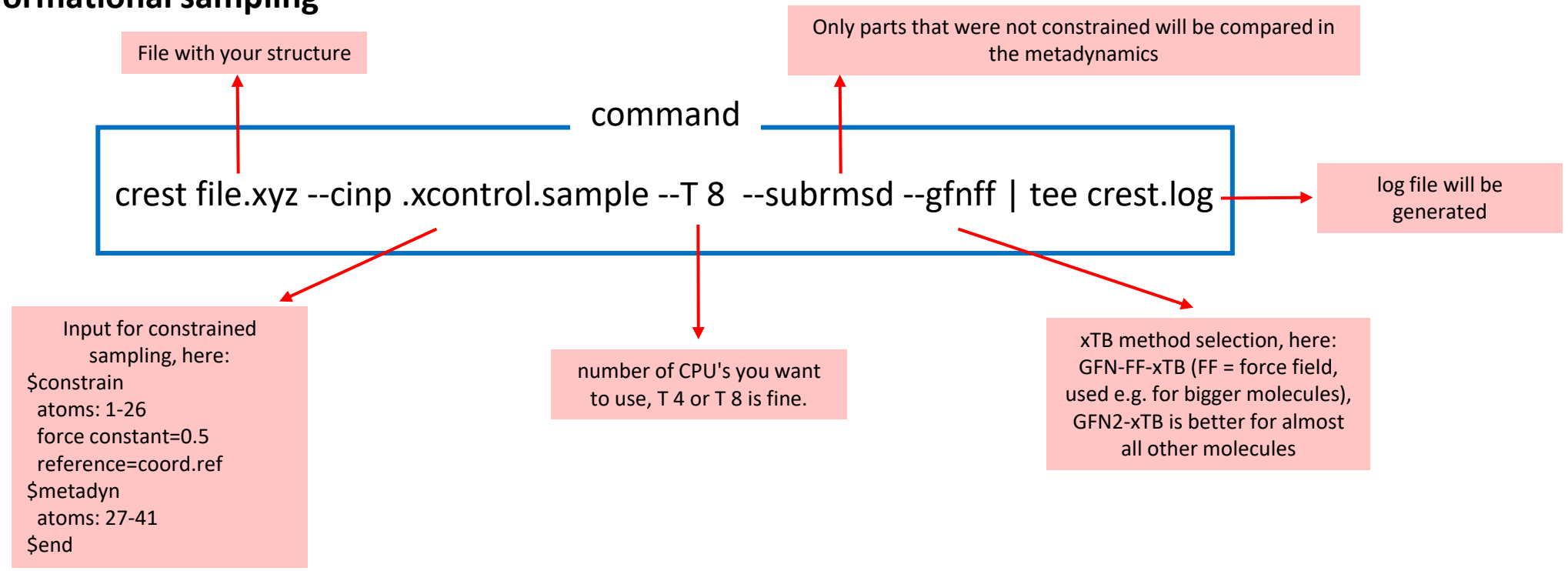


CREST

- CREST is using xTB (GFN2-xTB is default)
- Different tools:
  - Conformational Sampling to obtain all possible conformers and rotamers
  - Constrained conformational sampling if you want to make sure, that a specific criteria is met e.g. specific bond length, bond angle or dihedral angle should not change.
  - (De)protonation of a molecule to find the position of a molecule that will most likely be (de)protonated.
  - Quantum cluster growth (QCG) for an explicit solvation model to add the solvent molecules to your molecule to build up solvent shells for molecular dynamic simulations (MD)
- See documentation for the keyword you want/have to use

CREST

Constrained conformational sampling



- **.xcontrol.sample** and **coord.ref** files can be generated for you via crest: crest file.xyz --constrain 1-26

atoms you want to constrain

*more detailed explanation: https://crest-lab.github.io/crest-docs/page/examples/example_4.html

CREST

Constrained conformational sampling

output

- crest.log (in orca: orca.out)
- crest_best.xyz containes most stable structure
- crest_conformers.xyz only conformers
- crest_rotamers.xyz conformers *and* rotamers

```
T /K                                   :   298.15
E lowest                               :   -29.18570
ensemble average energy (kcal)         :    0.038
ensemble entropy (J/mol K, cal/mol K) :   66.163   15.813
ensemble free energy (kcal/mol)        :   -4.715
population of lowest in %              :   26.826
 number of unique conformers for further calc           262
 list of relative energies saved as "crest.energies"

 -----------------
 Wall Time Summary
 -----------------
             test MD wall time :         0h : 0m : 1s
                 MTD wall time :         0h :51m :19s
      multilevel OPT wall time :         2h :54m :15s
                  MD wall time :         0h :30m :35s
                  GC wall time :         0h :16m :57s
--------------------
Overall wall time  : 4h :37m : 4s

 CREST terminated normally.
```

*more detailed explanation: https://crest-lab.github.io/crest-docs/page/examples/example_4.html

Most common errors in Gaussian

Very often the “Error termination” message can be easily understood. Here are some common reasons for the error :

Incompatibility between the number of electrons and the spin multiplicity
This error is identified by Gaussian and can be read in the last few lines of the output file where the “Error termination” message is given. In particular, if the system you want to simulate contains an odd number of electrons, the spin multiplicity 2S+1 **cannot** be 1 because the system cannot be in a singlet state (with S=0). For instance, if you have one unpaired electron, the spin multiplicity is 2 because S=1/2.

Syntax error in the input file
At the beginning of the output file, Gaussian summarizes the input file. If already in the input file Gaussian has identified an issue, it will indicate it when rewriting the input lines. The syntax error will be indicated with the symbol ‘ just below the incorrect string.

Most common errors in Gaussian

Convergence issues
Sometimes it can happen that the program terminates with an error message because it could not converge the calculation.

To identify this issue, you can open the output file and search for the string “Converged”. If below **all** occurrences of the string “Converged?” you do not read **four** times the string “YES”, then it means that your calculation did not converge.
In this case, it is possible to restart the calculation from one of the last configurations, by creating a new input file and starting the same calculation but with a different initial geometry. To select the new geometry, search for one of the last occurrences of the string “Input orientation” in the output file and copy the coordinates of the atoms into the new input file. With this new input geometry, restart the calculation.

Linux Commands

ls [option(s)] [file(s)]

If you run **ls** without any additional parameters, the program will list the contents of the current directory in short form.

-l

Detailed list.

-a

Displays hidden files.

cp [option(s)] *sourcefile targetfile*

Copies sourcefile to targetfile.

-i

Waits for confirmation, if necessary, before an existing targetfile is overwritten.

-r

Copies recursively (includes subdirectories).

mv [option(s)] *sourcefile targetfile*

Copies sourcefile to targetfile then deletes the original sourcefile.

-b

Creates a backup copy of the sourcefile before moving

-i

Waits for confirmation, if necessary, before an existing targetfile is overwritten

rm [option(s)] *file(s)*

Removes the specified files from the file system. Directories are not removed by **rm** unless the option -r is used.

-r

Deletes any existing subdirectories.

-i

Waits for confirmation before deleting each file.

cd [options(s)] [directory]

Changes the current directory. **cd** without any parameters changes to the user's home directory.

mkdir [option(s)] *directoryname*

Creates a new directory.

rmdir [option(s)] *directoryname*

Deletes the specified directory, provided it is already empty.

gzip [parameters] *file(s)*

This program compresses the contents of files. Files compressed in this way are given the extension .gz and need to be uncompressed before they can be used.

Use **gunzip** to decompress the packed gzip files so they return to their original size.

Commands to Access File Contents

cat [option(s)] *file(s)*

The **cat** command displays the contents of a file, printing the entire contents to the screen without interruption.

-n

Numbers the output on the left margin.

less [option(s)] *file(s)*

This command can be used to browse the contents of the specified file. Scroll half a screen page up or down with **PgUp** and **PgDn** or a full screen page down with **Space**. Jump to the beginning or end of a file using **Home** and **End**. Press **Q** to exit the program.

grep [option(s)] *"searchstring" filenames*

The grep command finds a specific "searchstring" in the specified file(s). If the search string is found, the command displays the line in which the searchstring was found along with the file name.

-i

Ignores case.

-l

Only displays the names of the respective files, but not the text lines.

diff [option(s)] file1 file2

The **diff** command compares the contents of any two files. The output produced by the program lists all lines that do not match.

What was not mentioned (But probably will be filled to this presentation)

- ~~Composite methods~~
- ~~Aproximation for pure functionals like RI in details~~
- ~~Aproximation for HF or MP2 containing functionals like RIJCOSX was not mentioned at all~~
- Some history
- ~~Integration grids~~
- Calculation time scaling with number of atoms, basis set size, etc...
- Improve basis set slides
- Improve excited state calculations

Authors

Ann-Kathrin Rückert

Iulia Turcas

Tomáš Neveselý

Exercise!