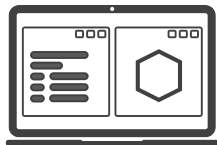


Introduction to Computational Chemistry

Part 4: Density Functional Theory

Eno Paenurk & Patrick Finkelstein & Felix Pultar



17.05.2023



- Part 1: Takeoff
- Part 2: Reaction pathways from scratch
- Part 3: Preparing structures
- **Part 4: Density Functional Theory**
- Part 5: Organometallics
- Part 6: Spectroscopy
- Part 7: Chemical concepts

What you should know from Parts 1-3



- Communicating with the cluster
- Generating xyz coordinates
- Running calculations with xtb



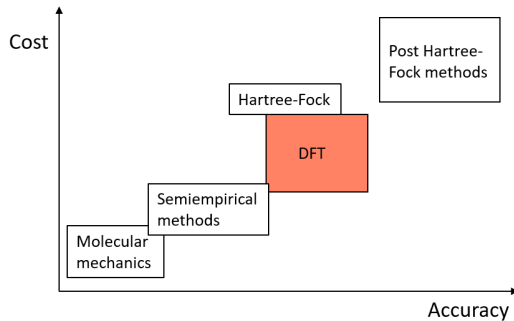
Theory:

- Density Functional Theory
- Introducing ORCA
- Functionals and basis sets
- Simple guidelines/problem-solving for calculations

Case study (combined with the live demo):

- Running common job types with DFT
- Extracting and visualizing the results

Recap: Different levels of theory



Disclaimer: This is merely a simplified representation



- Energy is expressed as a functional of the electron density
- Technically, the theory behind Kohn Sham DFT is exact. However, we don't know the exact expression—use approximations (functionals)
- SCF—an iterative algorithm to calculate the energy of a structure
- Issues:
 - London dispersion
 - “Multireference” character
 - “Functional zoo”



- *ab initio*, DFT, and semiempirical SCF-MO package
- Very user friendly
 - Simple input syntax
 - Human-readable output
 - 1000+ pages of detailed manual
 - Active forum
- Free for all academic users
- Supported on Euler and Grace, downloadable for Win/Mac/Linux



The minimum input we need:

- Method – How do we calculate the energy?
- Basis Set – How do we represent the electrons?
- Job type – What do we want to do?¹
- System description – Which atoms and where, how many electrons?

¹We will discuss job types in more detail in the second part of the presentation.



Example: The minimum input we need for a geometry optimization:

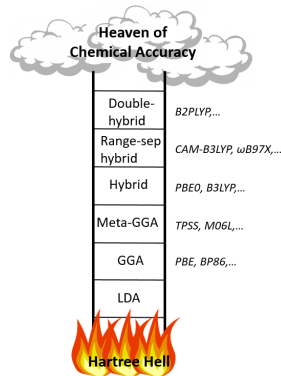
- **Method** – DFT with the BP86 functional
- **Basis Set** – def2-TZVP
- **Job type** – geometry optimization
- **System description** – charge 0, multiplicity 1, separate xyz file

Instead of running all of this on the command line we will now use explicit input files:

```
! BP86 def2-TZVP opt
*xyzfile 0 1 coord.xyz
```



- DFT uses functionals of the electron density to access the energy
- In ORCA there are over 60 pre-defined functionals to choose from.
- “Jacob’s Ladder”:



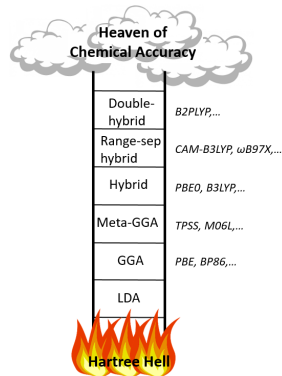


Which functional do I choose?

- Depends on what you want – geometry, property, etc.
- Depends on the system – size, transition metal, etc.
- Read benchmark studies
- Start with a cheap one, move on to expensive
- If unsure: try multiple (generally a good idea)

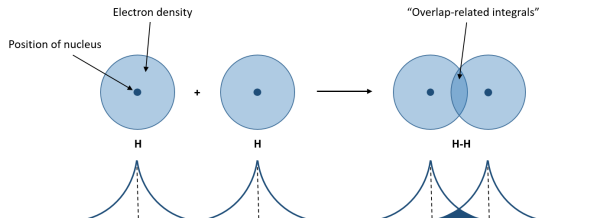


- BP86 for quick initial calculations
- PBE0 or B3LYP for more extensive ones
- Custom based on benchmark studies when the accuracy really matters





- Describe the molecular orbitals as a linear combination of atomic orbitals.
- Approximation: Use a linear combination of Gaussian-type functions.





- Basis set: A set of basis functions for each atomic orbital of an atom
- ζ (Zeta) classification: number of functions used to describe one atomic valence orbital, e.g., double-zeta, triple-zeta, etc.

Families of basis sets:

- Pople-type (6-311G, 3-21G, ...)
- Ahlrichs-Weigend (def2-SVP, def2-TZVP, ...)
- Correlation-consistent (cc-pVDZ, aug-cc-pVTZ, ...)
- Several more



What basis set do I choose?

- Careful with too small basis sets
- Always use polarized basis sets, e.g. def2-TZVP
- Does your compound have a negative charge? - Use diffuse basis sets, e.g., def2-TZVP
- Combinations are possible, e.g. larger basis set for heavier atoms, smaller basis set for lighter atoms



The def2 family is optimized for all elements (except 7th row), well balanced and efficient, and features intuitive control of the different levels:

- def2-SVP for quick initial calculations or very large systems
- def2-TZVP for production calculations
- def2-QZVPP for high accuracy single-points



- Provide charge and multiplicity ($= 2S+1$)
- Provide the coordinates in an external file (same folder, or give the path)

```
*xyzfile 0 1 coord.xyz
```

- Or we can write them directly in the input file

```
*xyz 0 1  
O 0.0 0.0 0.0  
H 0.0 1.0 0.0  
H 0.0 0.0 1.0  
*
```



So far:

- **Method** – DFT with the BP86 functional
- **Basis Set** – def2-TZVP
- **Job type** – geometry optimization
- **System description** – charge 0, multiplicity 1, separate xyz file

```
! BP86 def2-TZVP opt
*xyzfile 0 1 coord.xyz
```

However, still some things to add!



London dispersion:

- Grimme's correction: D3BJ

Solvation:

- Implicit: CPCM(solvent) or SMD (see Handout)

Example:

```
! BP86 def2-TZVP D3BJ CPCM(Toluene) opt
*xyzfile 0 1 coord.xyz
```



Add information about the resources we allow ORCA to use: number of cores and memory (RAM) per core.

Example for 2 cores with 1024 MB each:

```
! BP86 def2-TZVP opt
%pal nprocs 2 end
%maxcore 1024
*xyzfile 0 1 coord.xyz
```



- Method
- Basis Set
- Dispersion
- Solvent
- Job type
- Resources
- System description

```
! BP86 def2-TZVP D3BJ CPCM(Toluene) opt
%pal nprocs 2 end
%maxcore 1024
*xyzfile 0 1 coord.xyz
```



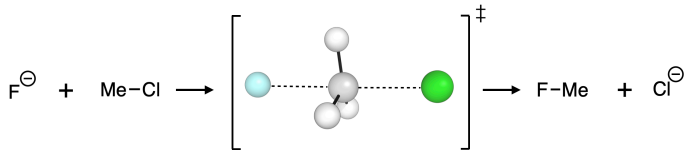
Additional keywords that can control accuracy:

- Tighter numerical criteria for optimization: e.g., `tightOpt`
- Tighter numerical criteria for SCF calculation: e.g., `tightSCF`

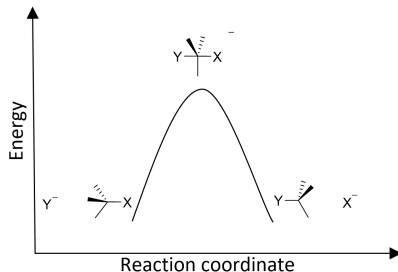


We will combine theory, recommendations, and the live demonstration to make a reaction energy profile for a classic Finkelstein reaction.

Our model system for this:



Caveat: We are ignoring a lot of things here, such as the salt formation, which tends to be a major driving force for the reaction.



- **Reactants and Intermediates:** Local minima on potential energy surface
- **Transition States:** First-order saddle points connecting two minima
- The energy is typically the Gibbs free energy (either kcal/mol or kJ/mol – careful with the units!)



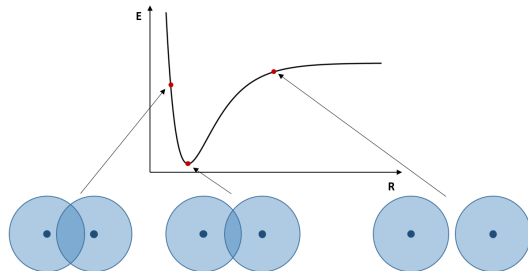
- **Geometry optimizations:**
 - **Groundstate:** `opt`
 - **Transition states:** `optts`
- **Frequencies:** `freq` or `numfreq`
- **Single point:** `sp` (default option)

Groundstate geometry optimization: Why do we do this?



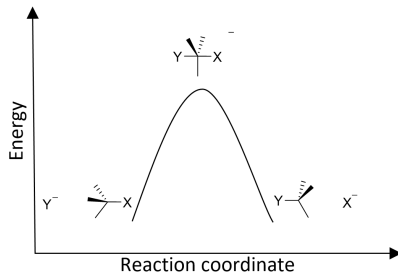
- Minimum energy structure should be the best physical representation of the system of interest
- All properties should thus be calculated on the minimum energy structure
- Thus, geometry optimizations should precede other calculations

Groundstate geometry optimization: How do we do this?



- 1 Calculate the energy of the starting structure
- 2 Introduce small change to the position of the nuclei
- 3 Calculate the energy of new structure
- 4 Repeat steps 2 and 3 until conditions are satisfied

Transition state geometry optimization



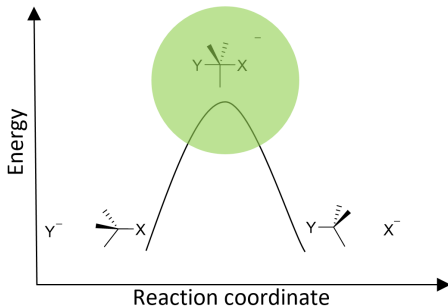
- Transition state = First order saddle point
- ORCA maximizes the energy along one vibrational mode (e.g., the bond breaking/being formed) while minimizing the energy for all others



Three commonly used strategies:

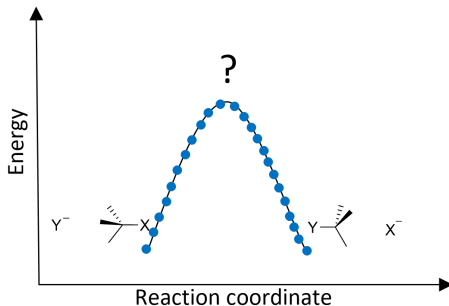
- 1 OptTS
- 2 Scan or ScanTS
- 3 NEB (Nudged Elastic Band)

Transition state geometry optimization: OptTS



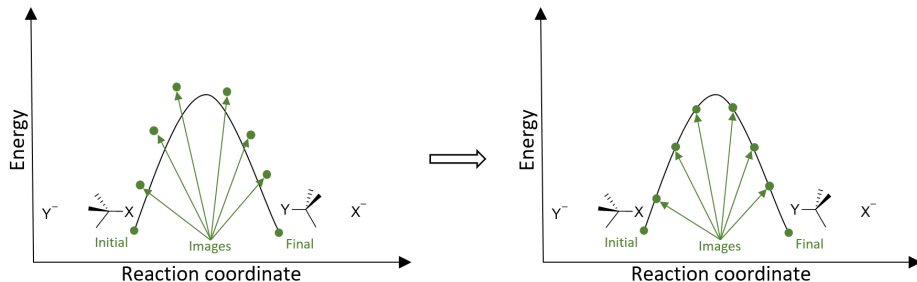
- Requires a (very) good guess for the transition state
- Common to employ a different strategy first and use OptTS to finalize
- Uses the keyword OptTs

Transition state geometry optimization: Scans



- Identify the key bond/angle and scan the energy w.r.t. changes in the key bond/angle
- ScanTS will perform a relaxed surface scan and the maximum energy structure will be used as input for a OptTS calculation (also possible to do only the first step using the scan keyword)

Transition state geometry optimization: NEB



- Initial and final system coordinates are required
 - Atom ordering must remain the same!
- Interpolation leads to a rough approximation of the reaction path – creation of “images”
- Correlated optimization of the images

Frequencies: Why do we need this?



- Calculates the normal mode frequencies from second derivatives (Hessian, .hess file)
- Vibrational spectra (See Week 6)
- Confirm ground state: 0 imaginary frequencies
- Confirm transition state: 1 imaginary frequency
- Calculate the zero-point energy for different isotopes – see KIEs
- Calculate Gibbs Free energy (or extract the correctional term)

Total Gibbs free energy $G_{tot} = E_{electronic} + H_{correction} - TS$

Frequencies: How do we do this?

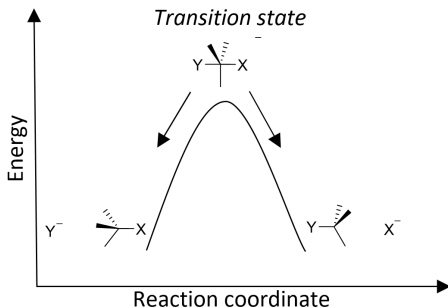


- Use the keyword `freq` (or `numfreq`)
- **Important:** Use the exact same method and basis sets as for the geometry optimization – otherwise the calculation is meaningless
- Commonly calculated directly after the geometry optimization, use combined keywords `opt freq` or `optts freq`



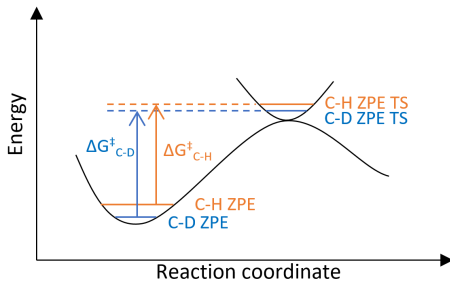
Two main methods to confirm transition states

- Frequency analysis: only one imaginary frequency for a transition state
- Intrinsic reaction coordinate (IRC): looks for the nearest intermediates – should correspond to both reactant and product





- Within the Born-Oppenheimer approximation, isotopes have the same PES, but different zero-point energies
- Obtain the KIE by calculating the activation energies for different isotopes²(see Handout)



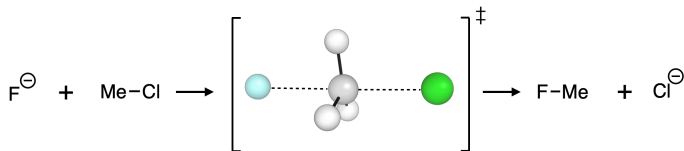
²Please note that there are more sophisticated and accurate methods to calculate KIEs

Single point: Why do we do this?



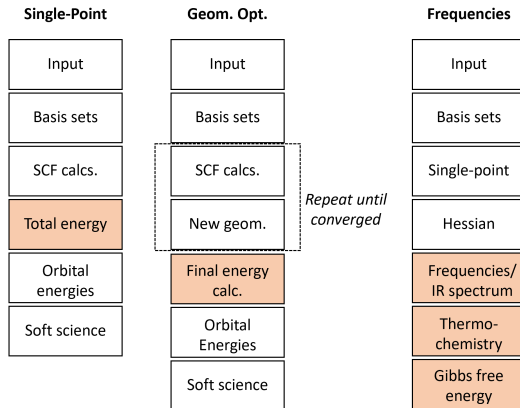
- Common to calculate the electronic energy of the optimized structures with a higher level of theory and/or larger basis sets
- Possible to extract properties that depend on the electronic energy and require more refined options (See Week 6 and 7)

Single point is the default job type and does not require a keyword (even so, there is one, `sp`)



- 1 Optimized groundstates and transition state and ran frequency calculations
 - Confirmed ground state and transition state
 - Extracted correction term
- 2 Ran more accurate single point calculation
- 3 What we would do next: Add the electronic energy to the correction term

The ORCA output file





Most properties can be viewed in an external viewer:

- Geometry/SCF convergence
- Energy values – careful with the units! (See Handout)
- All generated structures
- IR spectrum (will be discussed more in Week 6)



Common reasons why a job did not terminate correctly:

- Mistake in the input file, e.g., wrong keyword or typo
- Time limit – submitted to too short queue
- Memory limit – asked for too little memory
- SCF didn't converge

Common problems with correctly terminated jobs

- Too many imaginary frequencies
- Unexpected structural changes
- Issues with the electronic structure (See Week 5)



Where to look for error details:

- .err file
- .out file

Where to find help (all links provided in the Handout):

- ORCA manual
- ORCA forum
- ORCA input library

