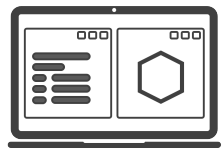
Introduction to Computational Chemistry

Part 1: Takeoff

Eno Paenurk & Patrick Finkelstein & Felix Pultar



26.04.2023

About Us

Dr. Eno Paenurk
(Hammes-Schiffer Group)

Patrick Finkelstein
(Morandi Group)

Dr. Felix Pultar
(Riniker Group)

Our computational background:

- Computational PostDoc, PhD, and MSc projects
- Computational chemistry workshops (attended and organized)
- Theoretical and computational chemistry courses at ETH

Our Motivation

Quantum and computational chemistry courses at ETH:

- Quantum Chemistry
- Advanced Quantum Chemistry
- Classical Simulation of (Bio)Molecular Systems

What organic chemists want:

- Practical Computational Chemistry

A recurring line when talking to organic chemists:
I would like to run some calculations but I don't know how :(

Our Philosophy

Provide a “from zero to (almost) hero” type of course.

What to expect

- A practical guide to typical calculations
- Brief discussions on advantages and limitations of the methods
- Hands-on exercises to learn running the calculations

What not to expect

- A theory/math-oriented introduction to quantum chemistry

End goal

Example: a JACS 2020 paper that cites B3LYP

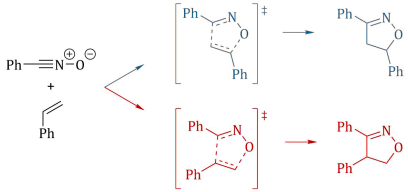
COMPUTATIONAL METHODS

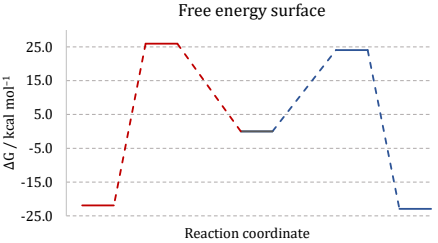
Density functional theory (DFT) geometry optimizations, energy, vibrational frequency, and coupled-cluster calculations were performed using ORCA version 4.1.2.[11] Geometry optimizations were carried out with the BP86 generalized gradient approximation (GGA) functional[12] with Becke-Johnson damped D3 dispersion corrections (herein referred to as D3).[13] A mixed basis set was implemented, in which palladium is described by the small core LANL2TZ(f) basis set with the Hay−Wadt effective core potential (ECP),[14] while the 6-31G(d) basis set was used on all other atoms. Key structures were optimized at various levels of theory and compared to crystallographic data to ensure that consistent results are obtained across multiple methodologies (see SI for details).

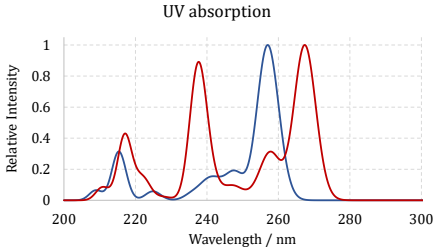
Triple-ζ quality single point calculations were carried out on all stationary points with a variety of density functionals, including BP86-D3,[12] B3LYP-D3,[15] PBE0-D3,[16] M06,[17] and DSD-BLYP-D3[18] (see SI for comprehensive list) with the def2-TZVP basis set[19] on all atoms (including the small core def2-ECP[20] on palladium). Corrections for solvation (THF or otherwise as specified) were carried out for single point calculations with the implicit Conductor-like Polarizable Continuum Model (CPCM).[21] Unless otherwise noted, all energies reported are G_{298} values from single point calculations at the M06/def2-TZVP−CPCM(THF) level of theory on BP86-D3/LANL2TZ(f)−6-31G(d) optimized geometries with thermodynamic corrections applied from frequency calculations obtained at the optimization level of theory. The Quasi-RRHO method is applied to correct for the breakdown of the harmonic oscillator approximation for low frequency vibrations.[22] All stationary points are characterized by the appropriate number of imaginary vibrational modes (zero for optimized geometries and one for transition states). Intrinsic reaction coordinate (IRC)

J. Am. Chem. Soc. **2020**, *142*, 13917.

End goal


Ph
N
O
Ph
+
Ph
‡
Ph
N
O
Ph
Ph
N
O
Ph
‡
Ph
N
O
Ph


Free energy surface
25.0
15.0
5.0
-5.0
-15.0
-25.0
ΔG / kcal mol⁻¹
Reaction coordinate


UV absorption
1
0.8
0.6
0.4
0.2
0
Relative Intensity
200
220
240
260
280
300
Wavelength / nm

Workshop overview

- 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

General structure

Overall schedule:

- 7 consecutive weeks
- Wednesdays from 5 pm until ∼7 pm

Workshop schedule:

- ∼45 min lecture + live demo
- ∼1 h exercise session

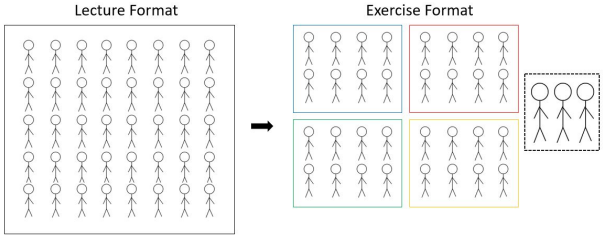
Associated content

Each session comes with 3 files that we will share via GitHub:

- Presentation – this very slideshow
- Handout – a short summary of important details + literature
- Exercise – a verbose walk-through of the exercises
- (Stuff – scripts, input templates, and other associated files)

Exercises: Format

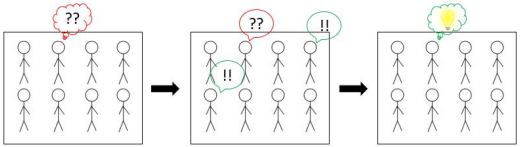
We will make a number of “Breakout Rooms”



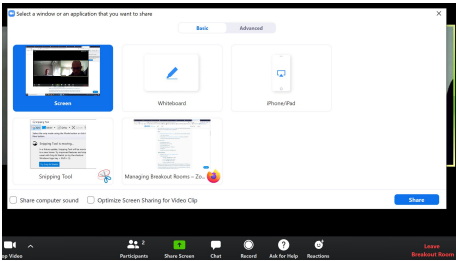
You can join a Breakout Room based on your affiliation or just pick one at random

Exercises: Problem resolving

Running into problems? Ask another participant

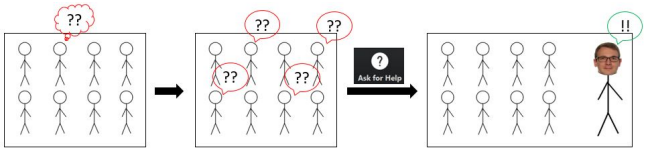


Share you screen with others to show/resolve problems

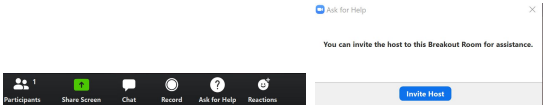


Exercises: Problem resolving

Running into problems and nobody can help or everyone has the same issue?



To call us you can press the “?” button and confirm



One of us will join as soon as we are free

Goals for today

- Accessing the cluster
- The essential bash commands
- File management
- Submitting jobs
- Optional: Getting familiar with the vim text editor

Computer clusters

- Access via the login node
- Submit calculations to the compute nodes
- Popular programs are usually pre-installed (accessed as modules)

Euler

The Euler cluster (I and II) in Lugano

- All ETH members have access (via VPN)
- 21 GB home directory (permanent)
- 2.7 TB scratch directory (temporary)
- Most QC programs available

Note

More information at: `https://scicomp.ethz.ch`

Important bash commands

The login node speaks bash – ergo, familiarity with basic bash commands is necessary. The absolute minimum:

Command	Description
`cd dummy`	changes the directory (folder) to "dummy"
`cd ..`	changes the directory to one level up
`ls` (or `ll`)	prints list of folders and files in current directory (with details)
`module load dummy`	loads the module (program + necessary software) called "dummy"
`sbatch --wrap="dummy"`	submits job of a command called "dummy"
`squeue`	prints the list of submitted jobs

Note

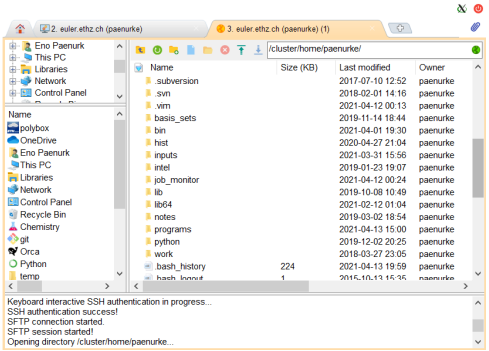
To print the output of a command into a file, use the > sign:

```
command > output
```

A more extensive list of useful bash commands is given in the Handout.

File handling

- Bash commands
- Graphical SFTP clients
  - FileZilla
  - MobaXterm
  - ...



Note

Text files generated in Windows have different line-endings than Linux text files. Use the `dos2unix` command to convert Windows files to Linux-compatible ones.

Running jobs

With the `sbatch` command you can:

- Submit *any* command or calculation to Euler
- Choose:
  - Number of CPUs
  - Amount of memory (determines the queue)
  - Maximum run time (determines the queue)

The cluster will:

- (Try to) run your command
- Write a summary of the errors in the `slurm` file

Note

Tool to generate the submission command:
`https://scicomp.ethz.ch/public/lsla/index2.html`

Live Demo

Exercises