

Introduction to Computational Quantum Chemistry

Lesson 05: Molecular Builders, Single Point Calculations and Geometry Optimizations



Some Available Molecular Builders on Wolf cluster

- Avogadro
 - + fast
 - + free
 - + forcefield pre-optimization
- Gabedit
 - + aligning structures in coordinate system
 - - relatively slow
- Gaussview
 - + generating specific distances/angles/dihedrals
 - + aligning molecules
 - - commercial

Structure Files

- the most widely used in QM are XYZ and Z-matrices
- Cube files:
 - grid of points with specific values of a given quality
 - electron density, Electrostatic potential, Laplacian of el. density.
 - generated from Gaussian wavefunction
- Wavefunction files:
 - Gaussian: (Formatted) Checkpoint Files
 - ADF: TAPE Files
 - Turbomole: MOS

Activity I: Exploring the Molecular Builders

- **NOTE: the Instructor will demonstrate the basics**
- *Open* **Avogadro**
 - drawing tools are located in the upper right side of the submenu bar, further specified modifications can be done in **Tool Settings** and **Display Settings**
 - performing preoptimization using **Auto Optimization Tool** is handy to ensure that the input molecule is at least classically in minimum before using to the main computational job
 - **Extensions** menu enables the user to generate input files of various computational softwares such as Gaussian as well as generating XYZ and Z-matrices formats
- explore the other builders such as **Gabedit** and **Gaussview**

Single Point Energy Calculations and Geometry Optimizations

- NOTE:
 - Single Point Energy Calculation is commonly default feature in the input file of any computational chemistry programs
 - Geometry Optimization Search contains series of SP runs thus commonly needed to be specified in the input file or in the execution command
- in this lecture we will be looking at **Gaussian**, **Turbomole** and **ADF** programs

Activity II: Gaussian

- prepare a Gaussian input file of a **small molecule of your choice** for Single Point Energy and Geometry Optimization calculations:
 - builders can generate the file, it can be written from scratch using **Avogadro**, **Gabedit**, **Gaussview**
 - common suffix is ".com"
 - use, 2 cores and at least 3 GB of memory
- the main Gaussian manual is available at: <https://gaussian.com/man/>, the slight variations of syntax from g09 to later versions are minimal, it is also specified in the website.

Activity II: the Gaussian Input (.com file)

- > a sample input file for single point energy calculation of formaldehyde in Z-matrix coordinates
- > # Route section which described the run type and basis set, adding **OPT** signifies geometry optimization
- > **0 1** signify charge and multiplicity
- > add Gaussian module then submit
- > *g09* input.com

```
%mem=6000000
%chk=../scratch/test1.chk
#P HF/6-31G(d) scf=tight

test1 HF/6-31G(d) sp formaldehyde

0 1
C1
O2 1 r2
H3 1 r3 2 a3
H4 1 r4 2 a4 3 d4

r2=1.20
r3=1.0
r4=1.0
a3=120.
a4=120.
d4=180.
```

- complete details of the input file is available at:
<https://gaussian.com/input/>



Activity II: Gaussian Log (.log file)

> the logfile of a job in **Normal Termination** (No Error) of Gaussian contains information about calculations cycles, SCF convergence, Energetics, etc
> for an **OPT** job, go to the end of the file and move backwards until you find the final set of forces and displacements, it should look something like this.

NOTE: the Instructor will run through the output file

```
Item          Value      Threshold  Converged?
Maximum Force .000090    .000450    YES
RMS Force     .000059    .000300    YES
Maximum Displacement .000352    .001800    YES
RMS Displacement .000230    .001200    YES
Predicted change in Energy=-4.580915E-08
Optimization completed.
  -- Stationary point found.

!      Optimized Parameters      !
!      (Angstroms and Degrees)   !
-----
! Name  Definition          Value      Derivative Info.
-----
! R1    R(1,2)              1.1945    -DE/DX = -0.0001
! R2    R(1,3)              1.1945    -DE/DX = -0.0001
! R3    R(1,4)              1.1945    -DE/DX = -0.0001
! A1    A(2,1,3)             120.0     -DE/DX = 0.0
! A2    A(2,1,4)             120.0     -DE/DX = 0.0
! A3    A(3,1,4)             120.0     -DE/DX = 0.0
! A4    L(2,1,4,3,-2)       180.0     -DE/DX = 0.0
-----
GradGradGradGradGradGradGradGradGradGradGradGradGradGradGradGradGradGradGradGradGrad
```

Gaussian Log (.log file)

- **Abnormal Termination** (with Errors) of Gaussian happens and troubleshooting can be tedious depending on the source of error. A list of common errors and how to correct them is available at this site:
https://docs.computecanada.ca/wiki/Gaussian_error_messages
- *qmutil*: nifty module to extract data from gaussian output:
 - *extract-gopt-ene* logfile
 - *extract-gopt-xyz* logfile
 - *extract-gdrv-ene* logfile
 - *extract-gdrv-xyz* logfile
 - *extract-xyz-str* xyzfile framenumbers
 - *extract-xyz-numstr* xyzfile

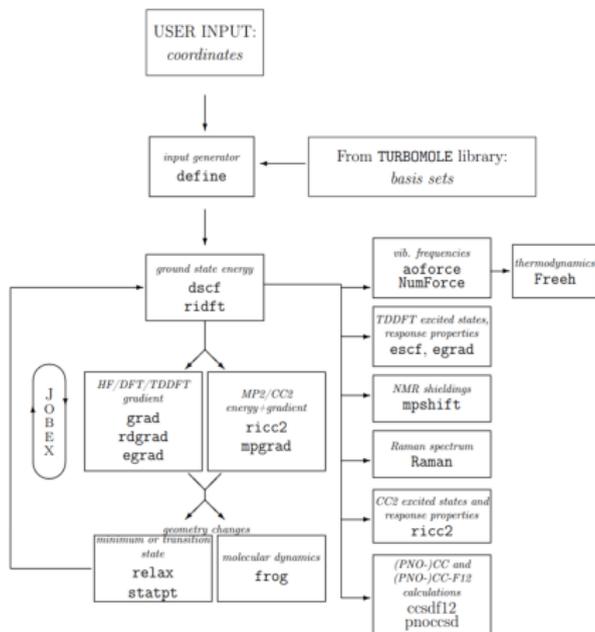
Gaussian Checkpoint (.chk file)

- stores wavefunction in binary
- convert into ASCII file:
 - *formchk* -3 file.chk
- can be read by (Gaussian GUI) Gaussview
- analysis of orbitals, electron density
- export into cubefiles (ASCII grid files)

Turbomole

- Turbomole is probably the “fastest” code available here (for Gaussian-type basis functions)
- Turbomole has interchangeable **Modules**, such that each contains everything necessary to execute only one aspect of the desired functionality.
- interactive preparation of the input file using *define*
- Turbomole manual: <https://www.turbomole.org/turbomole/turbomole-documentation/>

Modules of Turbomole, Main Data Flow



Activity III: Turbomole, preparing the job, *Define*

- add Turbomole module, then call *define*
- first two items can be skipped
- molecular geometry:
 - *a coord* Reads in the geometry
 - *ired* Generates internal coordinates
 - * Proceed to next stage
- basis set(s):
 - *b all def2-SVP* Assign this basis set to all atoms
 - * Proceed to next stage
- method
 - *eht* Perform initial guess from Extended Hückel Theory
 - *Accept all defaults*

Activity III: Preparing the job: Define *cont.*

- Method
 - **DFT** enter the DFT submenu
 - **on** use DFT
 - **func b-lyp** select the functional
 - **grid m5** increase the gridsize to m5
 - ***** exit the submenu
 - **ri** enter the RI submenu
 - **m** assign memory for RI
 - **2000** as much as possible
 - **on** use RI
 - ***** exit the submenu
 - **dsp** use dispersion correction
 - **on** use Grimme D3 correction
 - ***** exit the submenu
 - **marij** Multipole-Accelerated RI-J
- ***** end the define session

Activity III: Turbomole Job

- for running Turbomole in parallel mode use the parallel build
- infinity selects it by default if $n\text{cpu} > 1$

```
user@wolf
```

```
#!/bin/bash
```

```
module add turbomole:7.02:x86_64:para
```

```
dscf -ri -c 1024 > jobname.out
```

- remember **SP = "dscf"**, **OPT = "jobex"**,...
- it is safe to designate the results as a ".out" file

Activity III: Turbomole Output

- a successful Turbomole job creates several files

File	Contents
"name.out"	contains data for the chosen procedure
energy	Energies of steps
gradient	Gradients of steps
mos	Molecular orbitals
freq.out	Output from aoforce program

- to extract data for a geom.optimization run:

```
user@wolf
```

```
#!/bin/bash
```

```
module add turbomole
```

```
t2x > jobname.xyz (generates a series of geometry frames for the entire run)
```

```
t2x -c > jobname.xyz (generates the final geometry entire run)
```



ADF

- + the software is using Slater-type orbitals
- + up to 2-component relativistic effects (ZORA+SpinOrbit)
- + extremely fast and efficient
- + ADF GUI: adfview
- - "awful" output file structure
- - inefficient memory handling

Activity IV: Preparing ADF input

- keywords in blocks, case insensitive:

ATOMS*	definition of geometry in xyz
SYMMETRY NOSYMM	switch off all symmetry
XC*	DFT functional
BASIS*	Basis Sets Specifications
NOPRINT LOGFILE	do not print input into logfile
GEOMETRY* †	(added for geometry optimization)
RELATIVISTIC SpinOrbit(Scalar) ZORA	(if inclusion of relativistic effects)

† single point energy calculation is Default

* section terminated by **END** keyword

- MANUAL:

<https://www.scm.com/doc/Documentation/index.html>

Activity IV: ADF job

```
user@wolf
```

```
#!/bin/bash
```

```
module add adf
```

```
adf < input.inp > jobname.out
```

- it is safe to designate the results as a ".out" file

Activity IV: ADF Output

- important files of a succesful ADF run

File	Contents
"name.out"	contains data for the chosen procedure
TAPE* files logfile	contains various informations including wavefunction brief summary of the job

- using ADFVIEW (ADF GUI) for output visualization

```
user@wolf
#!/bin/bash
module add adf
adfview TAPE*
```

On the FINAL Note

- Gaussian
 - advisable for medium size molecules and if relativistic properties are not needed
- Turbomole
 - for large molecule optimizations
- ADF
 - if relativistic effects are needed

END