

PRACE Spring School in Computational Chemistry 2019

Quantum Chemistry Workshop (Advanced)

Excitation Energies and Excited State Densities

Mikael Johansson

<http://www.iki.fi/mpjohans>

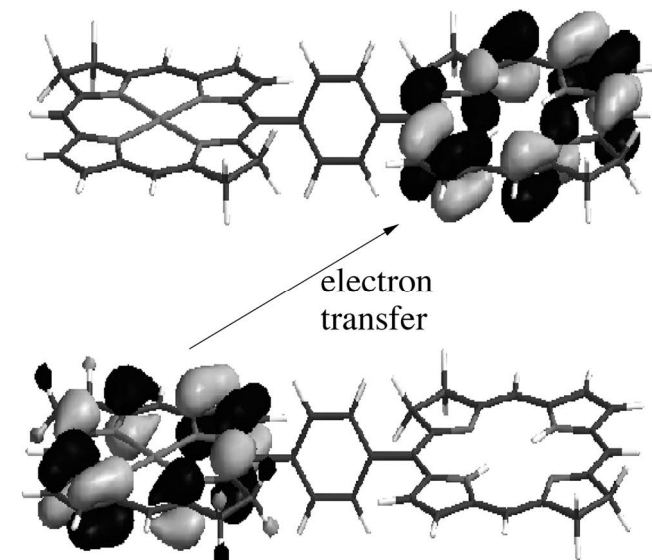
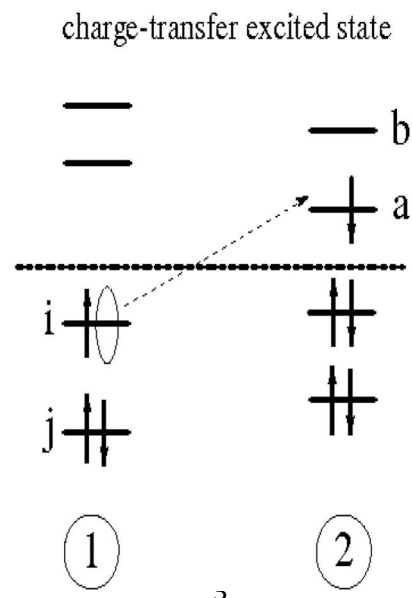
Objective: To learn how to compute electronic excitation energies at density functional theory and the coupled cluster levels using the NWChem program, as well as visualising how the total electron density is changed upon excitation

TD-DFT excitation energies

- Computing electronic excitation spectra (UV/VIS) is these days possible for large molecules due to efficient implementations of time-dependent density functional theory (TD-DFT). Despite the name, the most common usage of TD-DFT is to compute excited state properties within the linear response framework. In the case of excitation energies, no time dependence is actually modelled.
- Note that with excitation energies available, also excited state potential energy surfaces are available.
- With modern software (and hardware), TD-DFT calculations can routinely be performed for molecules with hundreds of atoms.
- The quality of excitation energies obtained by TD-DFT depends on the exchange-correlation functional used.
 - Relative excitation energies are reproduced better than absolute values
 - Hybrid functionals usually perform better compared to GGA functionals. The part of exact exchange that hybrids include help correct for the **wrong asymptotic behaviour** that most functionals suffer from. Long-range corrected functionals (LC-DFT) usually fare even better.
- Generally, the results obtained by TD-DFT have been found to be satisfactory, especially considering how computationally cheap they are to compute.

TD-DFT excitation energies

- There are some special categories of excitations that approximate functionals describe *quite poorly*:
 - Conical intersections at the excited PES (multi-reference treatment necessary)
 - Excitations that have double excitation character
 - Rydberg excited states
 - **Charge transfer excitations**
- In CT excitations, the incoming photon excites an electron which then moves over a large distance in the molecule. *Standard TDDFT severely underestimates these excitation energies!*
- For a good overview with examples of this, see for example A. Dreuw, M. Head-Gordon, *J. Am. Chem. Soc.* **126** (2004) 4007. <http://dx.doi.org/10.1021/ja039556n>



Self-interaction error / delocalisation error

- The Coulomb term in both **DFT** and **Hartree—Fock** contains an **unphysical interaction between an electron and itself**

$$J[\rho] = \frac{1}{2} \int d\vec{r}_1 \int d\vec{r}_2 \frac{\rho(\vec{r}_1)\rho(\vec{r}_2)}{r_{12}}$$

- With this unphysical interaction, the effective Hartree(-Fock)/KS potential *is the same for all orbitals*
- In HF, no problem, as this is **exactly cancelled** by the exchange term, when $i = j$

$$J[\rho] = +\frac{1}{2} \sum_{ij} \int d\vec{r}_1 \int d\vec{r}_2 \frac{\psi_i^*(\vec{r}_1)\psi_i(\vec{r}_1)\psi_j^*(\vec{r}_2)\psi_j(\vec{r}_2)}{r_{12}}$$
$$E_x^{\text{HF}} = -\frac{1}{2} \sum_{ij} \int d\vec{r}_1 \int d\vec{r}_2 \frac{\psi_i^*(\vec{r}_1)\psi_j(\vec{r}_1)\psi_j^*(\vec{r}_2)\psi_i(\vec{r}_2)}{r_{12}}$$

- In KS-DFT, with approximate functionals, the XC-energy usually *does not* cancel the Coulomb self-interaction!

One-electron SIE

- Consider the hydrogen atom with just one electron

$$E[\rho] = T_s[\rho] + \underbrace{J[\rho] + E_{xc}[\rho]} + E_{ne}[\rho]$$

These should cancel
for a one-electron system

| | LDA PW92 | PBE | LC-PBE | BP86 | B3LYP | TPSS | M06-2X | HF |
|-------------------|-----------------|-----------------|-----------------|------------------|------------------|------------------|-----------------|------------------|
| E(tot) | -0.478710 | -0.499989 | -0.494950 | -0.500323 | -0.502442 | -0.500235 | -0.499162 | -0.499999 |
| E(J) | <i>0.298388</i> | <i>0.306818</i> | <i>0.301881</i> | <i>0.306362</i> | <i>0.308328</i> | <i>0.310512</i> | <i>0.306999</i> | 0.312499 |
| E(xc) | -0.278121 | -0.307457 | -0.297570 | -0.307144 | -0.311174 | -0.311018 | -0.307387 | -0.312499 |
| SIE (a.u.) | 0.020267 | -0.000639 | 0.004311 | -0.000783 | -0.002846 | -0.000505 | -0.000388 | 0 |
| (kcal/mol) | 12.7 | -0.4 | 2.7 | -0.5 | -1.8 | -0.3 | -0.2 | 0 |

- Almost all approximate functionals suffer from self-interaction error!**
 - Note energies below the correct answer of -0.5 Hartree
 - Note different Coulomb energies between methods: different densities!
 - To minimise Coulombic self-repulsion, delocalised orbitals are unjustly favoured
 - Too diffuse orbitals with SIE**
- SIE present in both exchange and correlation functionals

Insights into Current Limitations of Density Functional Theory

Aron J. Cohen, Paula Mori-Sánchez, Weitao Yang*

Density functional theory of electronic structure is widely and successfully applied in simulations throughout engineering and sciences. However, for many predicted properties, there are spectacular failures that can be traced to the delocalization error and static correlation error of commonly used approximations. These errors can be characterized and understood through the perspective of fractional charges and fractional spins introduced recently. Reducing these errors will open new frontiers for applications of density functional theory.

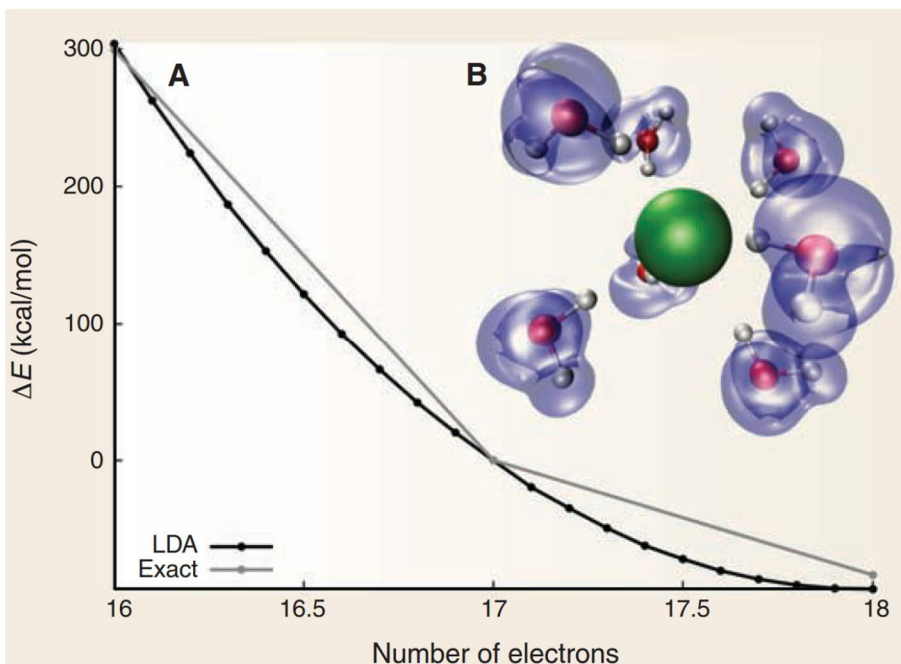
Science **321** (2008) 792,

<http://dx.doi.org/10.1126/science.1158722>

- The energy should change linearly between N and $N+1$ electrons: **derivative discontinuity**
- Instead, **convex**, with **too low energy** for partial electrons
- Problematic with subsystems within the whole
- **Wrong dissociation of radicals** (odd total electrons!)

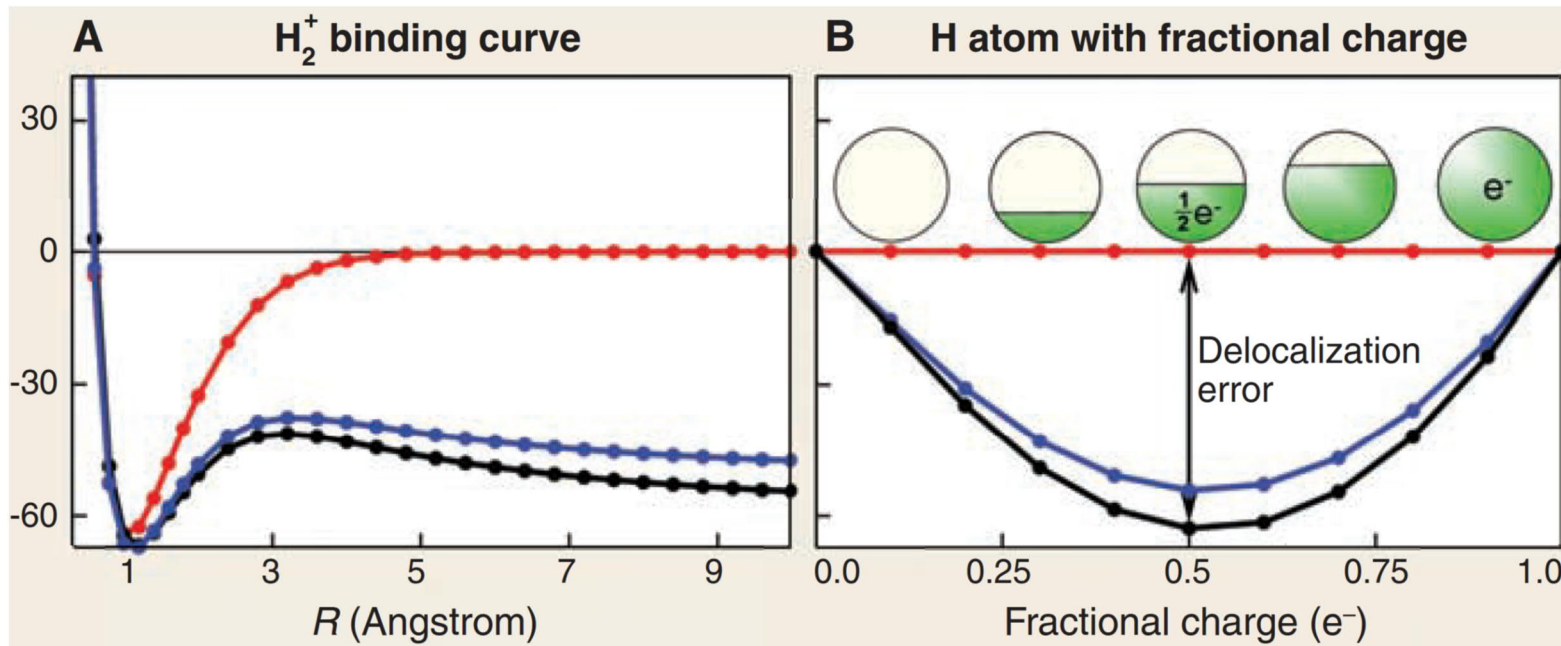
Delocalisation error

- Wrong behaviour of the energy of systems with **partial number of electrons**



Dissociation of H_2^+

- A rather embarrassing case for DFT: **qualitatively wrong dissociation of the 1-electron system H_2^+**
 - Energies in kcal/mol:



- In general, always a problem with dissociating radical ions!

Other practical consequences of SIE in DFT

Occupied orbital energies are too high

- IPs from HOMO too low
- Band gap too small
- Sometimes, aufbau principle *cannot* be satisfied!
- Problems with anions, HOMO orbital energy can become positive, that is, unbound!

Self-interaction introduces nondynamic (left–right) correlation

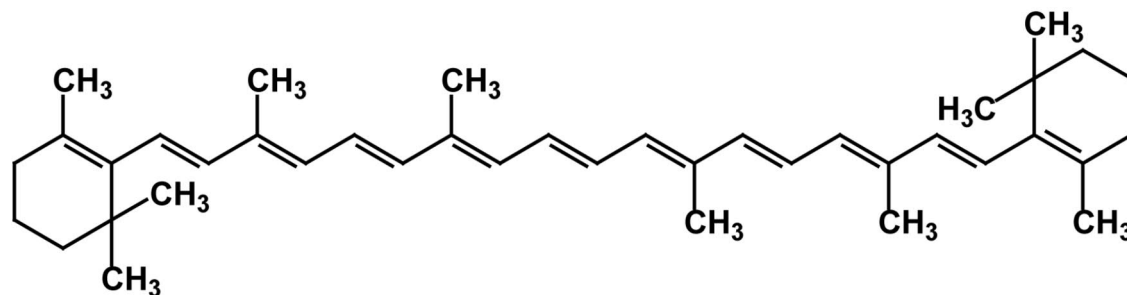
- Exchange functionals in DFT have been found to introduce correlation that in WF terminology “should” not be treated by exchange
- Mild multireference systems can for this reason be treated quite well with DFT, much better than with lower order WF correlation methods!
- SIE-corrected exchange functionals seem to include *no* electron correlation

More problems caused by SIE

- **Reaction barriers** can become too low, especially for atom-transfer reactions
- **Wrong asymptotic behaviour** of the XC potential
 - Approximate functionals vanish exponentially (too fast) instead of:
 - **Negative ions** often unbound
 - Occupied orbital energies too high

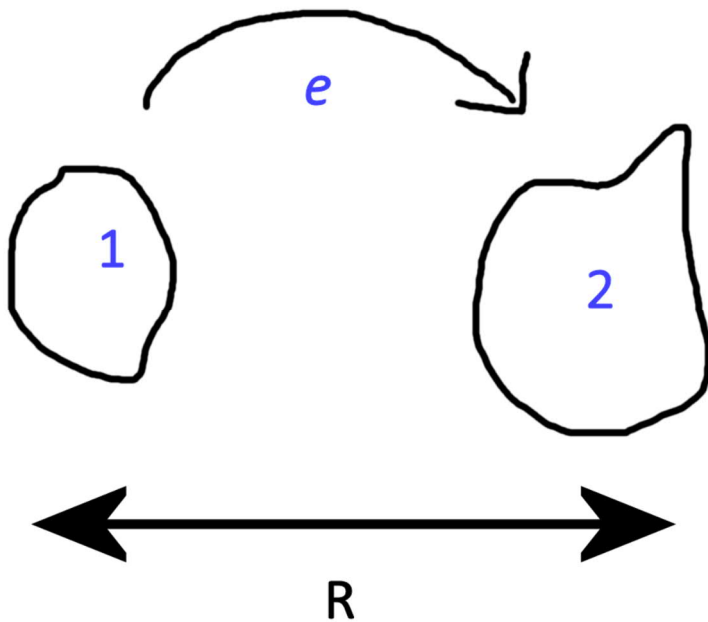
$$\lim_{r \rightarrow \infty} v_{\text{xc}}(\mathbf{r}) = -\frac{1}{r} + I + \epsilon_{\text{HOMO}}$$

- **Polarizabilities** of long chains overestimated (conjugated π -systems, ...)



TD-DFT excitation energies

- The problems with **Rydberg excited states** and **charge transfer excitations** can be traced to the **self-interaction error** and the **wrong asymptotic behaviour of the XC potential**



When $R \rightarrow \infty$, the excitation energy of a charge transfer excitation should be $E(\text{CT}) = \text{IP}_1 - \text{EA}_2 - 1/R$,
IP = ionization potential
EA = electron affinity

Due to SIE, the IP is too low \rightarrow the excitation energy too low

The excitation energy computed by DFT is exactly the energy difference between the orbitals involved in the CT transition!

- In molecules, the XC potential of (semi)local functionals decays **exponentially**
 - The **exact** XC potential should decay as $-1/R$
- Hybrid functionals** a bit better, decay as $-c_{\text{HF}}/R$, where c_{HF} is the amount of exact HF exchange

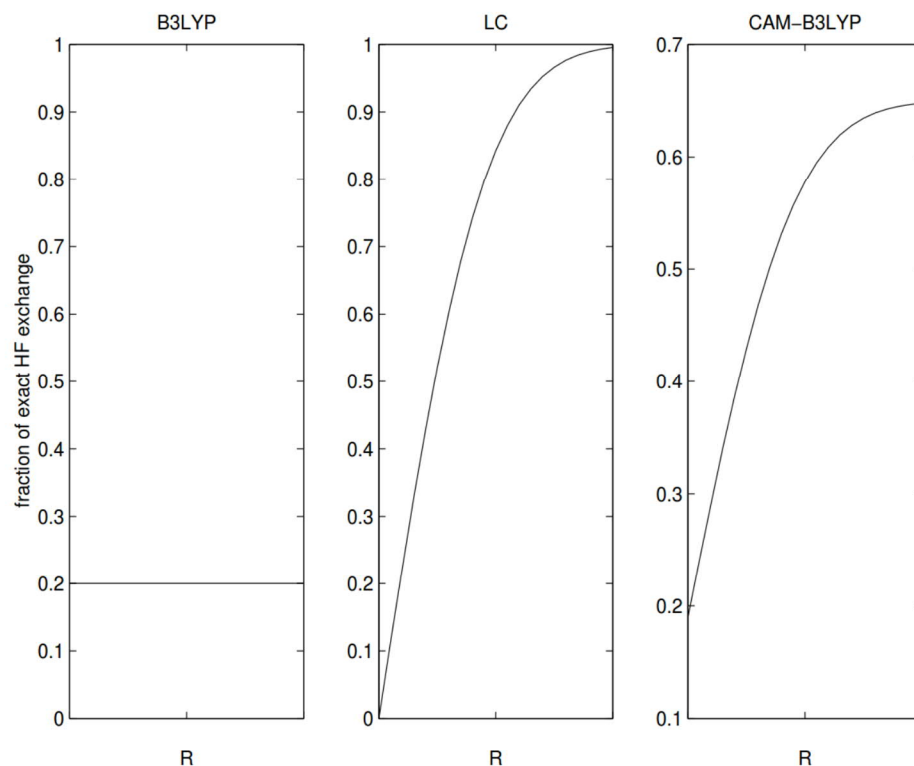
Long-range corrected functionals

- So, hybrid functionals improve the form of the XC potential at large R (as $-c_{\text{HF}}/R$)
- If c_{HF} would be 1 (pure HF exchange) the potential *would* be correct at large R
 - Pure HF exchange not a good idea, as discussed previously
- **How about changing the amount of HF exchange depending on the inter-electronic distance?**
 - This gives the **long-range corrected** version of DFT, **LC-DFT**
 - Also: **range-separated** functionals
- LC functionals partitions exchange into **short and long-range** terms smoothly using an error function
- **Short-range exchange** described mostly by the “normal” functional (typically 0—25 % HF exchange)
- **Long-range exchange** described mostly by exact HF exchange (typically 65—100 %)

Long-range corrected functionals

- LC functionals partitions exchange into **short and long-range** terms smoothly using an error function
 - the **attenuation parameter** μ (sometimes ω) defines how soon full HF exchange takes over (typically 0.3—0.5); usually fixed, better results when dynamic

$$\frac{1}{r_{12}} = \frac{1 - \text{erf}(\mu r_{12})}{r_{12}} + \frac{\text{erf}(\mu r_{12})}{r_{12}}$$



- Leads to much improved CT excitation energies, among other things

Coupled Cluster excitation energies

- The singles and approximate doubles coupled cluster model **CC2** is an alternative to TDDFT for computing excitation energies on large systems.
- While not (yet) as fast as TDDFT, quite large systems can already be studied. Of the wave-function based methods of computing excitation energies, it has, arguably, the best price/performance ratio.
 - It *can* describe charge transfer excitations
 - Still problems with double excitations, but you can get a diagnostic telling you if it's a problem!
- For a bit more background and an illustrative example, see C. Hättig, A. Hellweg, A. Köhn, "Intramolecular Charge-Transfer Mechanism in Quinolidines: The Role of the Amino Twist Angle", *J. Am. Chem. Soc.* **128** (2006) 15672. <http://dx.doi.org/10.1021/ja0642010>
- **Today** we will compute excitation energies with the full CCSD model, and at least in principle learn how to compute also CCSDT excitation energies

Without further ado, let's go through the basic usage of **NWChem**, the program of choice today.

The NWChem Program — An Introduction

NWChem can compute a lot of stuff at HF, DFT, and correlated wave-function theory levels: Excitation energies, vibrational spectra, excited state geometries, *etc.* It has, from the outset, been written with scalability for supercomputers in mind.

Web page:

- <http://www.nwchem-sw.org/>
 - Manual, downloads, a user forum
 - <https://github.com/nwchemgit/nwchem/wiki>
 - New pages from version 6.8 onwards
-
- Here, we will not really go through how to use NWChem in general, but rather focus on how to get the program to compute excitation energies at DFT and CC levels.
 - The input structure is quite clear
 - There are, however, plenty of undocumented or poorly documented options

Setting up the environment

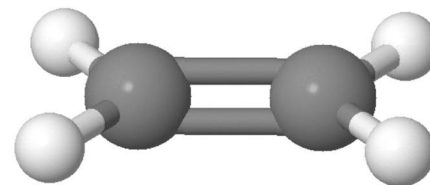
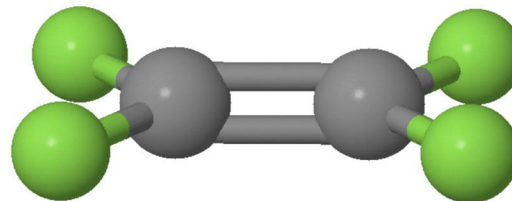
- We will run NWChem on Sisu: <https://research.csc.fi/sisu-user-guide>
- You should have (received) a user name for the machine, use SSH or something else to login
- We will use the test or short queue to run our jobs; for today, a general jobfile like this should suffice:

```
#!/bin/sh
#SBATCH --job-name=excited
#SBATCH --output=jobfile.out
#SBATCH --error=jobfile.err
#SBATCH -N 4
#SBATCH --time=00:30:00
#SBATCH --partition=test
#SBATCH --no-requeue
SDIR=`pwd`
echo "Submission directory is: $SDIR"
echo "The job ID assigned by the Batch system is: $SLURM_JOBID"
module unload PrgEnv-cray
module unload PrgEnv-intel
module unload PrgEnv-gnu
module load PrgEnv-intel
module load nwchem/6.8
(( ncores = SLURM_NNODES * 24 ))
aprun -n $ncores -N 24 -S 12 -ss nwchem excited.nw > excited.out
```

- Submit using `sbatch job-nwchem.job`, after creating the `excited.nw` input file

Setting up the calculations

- Let's first go through a few input examples “interactively”, run them, and continue with the actual exercises
- Our example system today will be the ethene/fluoroethene “dimer”, separated by an appreciable distance
 - A bit artificial, but contains a prototypical **charge-transfer excitation between the HOMO and LUMO orbitals!**
- Let's test how these three levels of theory describe the excitations in this system:
 - Standard DFT
 - Long-range corrected DFT
 - CCSD



Example 1: TD-DFT energies at PBE/def2-SVP level

START etfet

TITLE "EtFet charge transfer"

memory total 2000 mb

GEOMETRY

| | | | |
|---|-------------|-------------|-------------|
| C | 0.00000000 | -0.67046000 | -5.71886438 |
| C | 0.00000000 | 0.67046000 | -5.71886438 |
| H | 0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | 1.25053000 | -5.71913438 |
| H | 0.93766000 | 1.25053000 | -5.71913438 |
| C | 0.00000000 | -0.67241000 | 1.90626563 |
| C | 0.00000000 | 0.67241000 | 1.90626563 |
| F | 1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | 1.39635000 | 1.90632563 |
| F | 1.10887000 | 1.39635000 | 1.90632563 |

Symmetry C2v

END

BASIS **spherical**

* library def2-SVP

END

basis "cd basis" spherical

* library "Weigend Coulomb Fitting"

end

```
SCF
  direct
End
```

```
DFT
  xc xpbe96 cpbe96
  Convergence Energy 1e-8
  Vectors output etfet.movecs
END
```

```
TDDFT
# RPA
  CIS
  NROOTS 10
  SINGLET
  NOTRIPLET
  TARGETSYM b2
  SYMMETRY b2
END
```

```
TASK TDDFT ENERGY
```

Example 2: TD-DFT energy gradients at PBE/def2-SVP level

- For getting excited state densities (next example) we need to compute the gradients of the specific excited state in question
- Note: Only for one specified target excited state at a time
- Note: With gradients, one can also **optimise the excited state geometry** by changing
 - TASK TDDFT GRADIENT → TASK TDDFT OPTIMIZE
 - Optimising excited state geometries is *much* more complex than optimising GS structures, however
 - One difficulty is that one has to specify which excited state to optimise, *but the energy order of the excited states can (and often will) change* during the optimisation as the structure changes!

START etfet
TITLE "EtFet charge transfer"
memory total 2000 mb

GEOMETRY

| | | | |
|---|-------------|-------------|-------------|
| C | 0.00000000 | -0.67046000 | -5.71886438 |
| C | 0.00000000 | 0.67046000 | -5.71886438 |
| H | 0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | 1.25053000 | -5.71913438 |
| H | 0.93766000 | 1.25053000 | -5.71913438 |
| C | 0.00000000 | -0.67241000 | 1.90626563 |
| C | 0.00000000 | 0.67241000 | 1.90626563 |
| F | 1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | 1.39635000 | 1.90632563 |
| F | 1.10887000 | 1.39635000 | 1.90632563 |

Symmetry C2v

END

BASIS spherical

* library def2-SVP

END

basis "cd basis" spherical

* library "Weigend Coulomb Fitting"

end

```
SCF
  direct
End
```

```
DFT
  xc xpbe96 cpbe96
  Convergence Energy 1e-8
  Vectors output etfet.movecs
END
```

```
TDDFT
# RPA
  CIS
  NROOTS 2
  SINGLET
  NOTRIPLET
  CIVECS
  GRAD
  ROOT 1
  END
  TARGETSYM b2
  SYMMETRY b2
END
```

```
TASK TDDFT GRADIENT
```

Example 3: Visualising the movement of the electron upon excitement

- The DPLOT module takes care of preparing grid based 3D data
- We will use it to output Gaussian CUBE file format, as this format is widely recognised by many visualisation programs
- To get the electron density difference between the ground state and the excited state, we need to make two cube files, one with the GS density and the other with the ES density, and then take their difference
- To get NWChem to create the cube files, we need a few additional directives to the end of Example 2

Example 3a: Ground state density

```
dplot
  TITLE Ground State Density
  vectors etfet.movecs
  spin total
  LimitXYZ
  -2.5 2.5 50
  -2.0 3.0 50
  -8.0 4.0 120
  gaussian
  output gsdens.cube
End
task dplot
```

Example 3b: Excited state density

```
dplot
TITLE Excited State Density
densmat etfet.dmat
LimitXYZ
-2.5 2.5 50
-2.0 3.0 50
-8.0 4.0 120
gaussian
output exdens-1.cube
end
task dplot
```

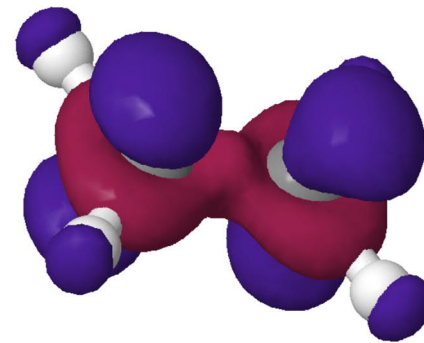
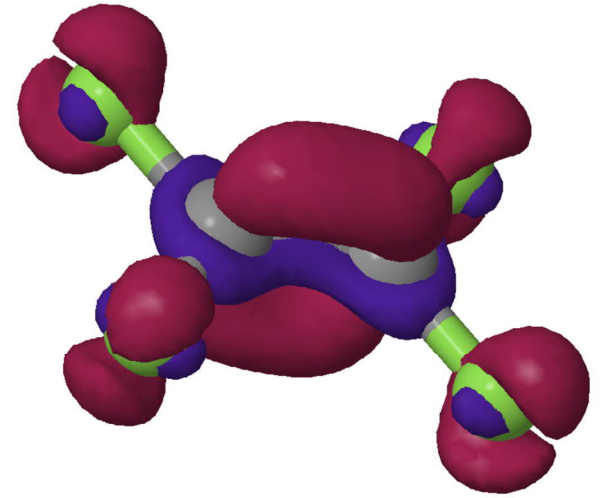
- There's many tools that can manipulate CUBE files. Let's use one that works well for simple arithmetics, CubeDiff, by Iñaki Silanes
`wget http://isilanes.org/soft/cubediff/cubediff.r97`
`chmod 755 cubediff.r97`
- `./cubediff.r97 +exdens-1.cube -gsdens.cube > ct.cube`

Example 3: Visualising the movement of the electron upon excitement

- Then let's load the resulting CUBE file with Jmol and look at the transition
 - Jmol ct.cube

Then, in the Jmol script window:

- isosurface pos cutoff 0.01 ct.cube
- isosurface neg cutoff -0.01 ct.cube



Example 4: Computing excitation energies using EOM-CCSD

- EOM-CCSD calculations are rather memory intensive, so we will **increase the number of nodes to 8** in order to not run out of memory
- Another parameter that **heavily influences performance and memory requirements** is the **tilesize** parameter of the TCE block
 - The higher the value, the faster the calculation (usually), but the more memory the calculation will require!
 - The related parameter `attilesize` should be larger than `tilesize`
- EOM-CCSD is done with the Tensor Contraction Engine (TCE) module of NWChem
- Some comments on some parameters:
 - **eomsol 2** Faster and less memory intensive than the default solver `eomsol 1` *but doesn't really work for double excitations* (uses CIS for initial guess).
 - **thresh 1.0e-5 / set tce:thresheom 1.0d-4**
CCSD ground state equations converged to 10^{-5} (*quite sloppy!*) but the EOM excitation energies only to 10^{-4} . The latter setting only works with `eomsol 2`.
 - **DIIS 8** DIIS iterations for (excitation) amplitudes: Should be ≥ 5 ; sometimes playing around with this avoids convergence problems, *e.g.*, situations with $E_{exc} \rightarrow 0$
 - Plenty of other parameters to tweak and tune in the TCE!

START etfet

TITLE "EtFet charge transfer"

memory stack 800 mb heap 200 mb global 1000 mb

GEOMETRY

| | | | |
|---|-------------|-------------|-------------|
| C | 0.00000000 | -0.67046000 | -5.71886438 |
| C | 0.00000000 | 0.67046000 | -5.71886438 |
| H | 0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | 1.25053000 | -5.71913438 |
| H | 0.93766000 | 1.25053000 | -5.71913438 |
| C | 0.00000000 | -0.67241000 | 1.90626563 |
| C | 0.00000000 | 0.67241000 | 1.90626563 |
| F | 1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | 1.39635000 | 1.90632563 |
| F | 1.10887000 | 1.39635000 | 1.90632563 |

Symmetry C2v

END

BASIS spherical

* library def2-SVP

END

```
SCF
  SINGLET
  RHF
  thresh 1.0e-10
  tol2e 1.0e-10
END
```

```
TCE
  CCSD
  2orb
  eomsol 2
  diis 8
  tilesize 40
  attilesize 50
  DIPOLE
  FREEZE CORE ATOMIC
  NROOTS 1
  thresh 1.0e-5
END
```

```
set tce:thresheom 1.0d-4
```

```
TASK TCE ENERGY
```

Example 5: Computing electron densities at CCSD level

- This functionality is still quite preliminary, and almost undocumented
- Some apparent limitations:
 - Only works without symmetry (Symmetry C1)
 - Frozen orbitals cannot be used
 - Does not work with 2EORB two-electron integral scheme (= needs much more memory!)
- Let's have a look anyway

Example 5a: Computing the ground state CCSD density

START etfet

TITLE "EtFet charge transfer"

memory stack 800 mb heap 200 mb global 1000 mb

GEOMETRY

| | | | |
|---|-------------|-------------|-------------|
| C | 0.00000000 | -0.67046000 | -5.71886438 |
| C | 0.00000000 | 0.67046000 | -5.71886438 |
| H | 0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | 1.25053000 | -5.71913438 |
| H | 0.93766000 | 1.25053000 | -5.71913438 |
| C | 0.00000000 | -0.67241000 | 1.90626563 |
| C | 0.00000000 | 0.67241000 | 1.90626563 |
| F | 1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | 1.39635000 | 1.90632563 |
| F | 1.10887000 | 1.39635000 | 1.90632563 |

Symmetry C1

END

BASIS

* library 6-31G

END

```
SCF
  SINGLET
  RHF
  thresh 1.0e-10
  tol2e 1.0e-10
END
```

```
TCE
  CCSD
  diis 5
  tilesize 30
  attilesize 40
  thresh 1.0e-4
  densmat etfet.densmat
END
TASK TCE ENERGY
```

```
DPLOT
  TITLE Ground State Density
  densmat etfet.densmat
  LimitXYZ
    -2.5 2.5 50
    -2.0 3.0 50
    -8.0 4.0 120
  gaussian
  output gsdens.cube
END
TASK DPLOT
```

Example 5b: Compute an excited state CCSD density

START etfet

TITLE "EtFet charge transfer"

memory stack 800 mb heap 200 mb global 1000 mb

GEOMETRY

| | | | |
|---|-------------|-------------|-------------|
| C | 0.00000000 | -0.67046000 | -5.71886438 |
| C | 0.00000000 | 0.67046000 | -5.71886438 |
| H | 0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | -1.25053000 | -5.71913438 |
| H | -0.93766000 | 1.25053000 | -5.71913438 |
| H | 0.93766000 | 1.25053000 | -5.71913438 |
| C | 0.00000000 | -0.67241000 | 1.90626563 |
| C | 0.00000000 | 0.67241000 | 1.90626563 |
| F | 1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | -1.39635000 | 1.90632563 |
| F | -1.10887000 | 1.39635000 | 1.90632563 |
| F | 1.10887000 | 1.39635000 | 1.90632563 |

Symmetry C1

END

BASIS

* library 6-31G

END

```
SCF
  SINGLET
  RHF
  thresh 1.0e-10
  tol2e 1.0e-10
END

TCE
  CCSD
  eomsol 2
  diis 8
  tilesize 30
  attilesize 40
  NROOTS 2
  thresh 1.0e-4
  densmat etfet.densmat
END
TASK TCE ENERGY

DPLOT
  TITLE Excited State Density
  densmat etfet.densmat
  LimitXYZ
    -2.5 2.5 50
    -2.0 3.0 50
    -8.0 4.0 120
  gaussian
  output exdens-2.cube
END
TASK DPLOT
```


Exercises, time allowing

1. Compute the excitation energy of the “HOMO-LUMO” CT transition using the LC- ω PBEh functional
2. Compare the excitation energy of this transition with the energies of the DFT HOMO and LUMO orbitals. What do you observe?
3. Plot the HOMO and LUMO orbitals using appropriate DPLOT commands

Some additional remarks

- If you need more than the default 2.6 GB/core for you NWChem calcs on SisU, you need to edit your job script to use only **a fraction of the cores per node**. For around 5 GB/core, use for example:

```
(( ncores = SLURM_NNODES * 12 ))  
aprun -n $ncores -N 12 -S 6 -ss nwchem excited.nw > excited.out
```
- For large TCE calcs (larger systems, larger basis sets), more cores can help if memory is limiting
- There’s also plenty of things to tune with the memory parameters. Beyond this tutorial, but adding these to the TCE block is a good start (uses more disk instead of in-core storage):

```
2eorb  
2emet 6  
idiskx 1  
split 2
```