Lecture 3:
Overview of Siesta

Oswaldo Diéguez

Department of Materials Science and Engineering
Tel Aviv University

Siesta and TranSiesta Tutorial, Tel Aviv, September 2014

ABOUT THIS LECTURE

- This is a very basic introduction to the Siesta code

- This lecture will serve also to prepare for more advanced hands-on exercises

Your Computer

- You are seating next to a computer operated by the Microsoft Windows operating system…

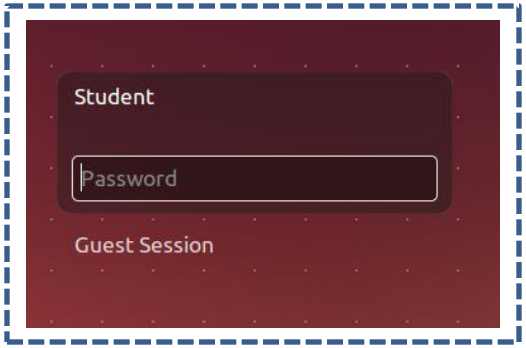
- …but for running Siesta, a Unix-based operating system is more effective; we want to use Ubuntu Linux…

- …so we are running a virtual Ubuntu machine in your host thanks to VirtualBox

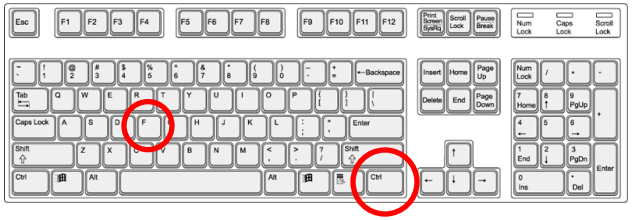
How to Run VirtualBox?

(1) Open VirtualBox

(2) Start the SiestaTutorialClone machine



(3) Login into the Ubuntu account;
password is **siesta2014**



(4) Go full screen mode: Right-Ctrl + F

Get a Copy of Siesta

1. Point a internet browser to: **www.icmab.es/siesta**
2. Visit **Access to the Code**
3. Visit the **Academic Licence**
4. Go **Back**; click in the first instance of **By Clicking I Agree**
   (if you intend to use Siesta in a commercial or non-academic environment, or as the representative of a computer center, you will have to obtain a different licence to work outside this classroom)
5. Click **Downloads of selected versions of Siesta**
6. Click on **siesta-trunk-462 (May 29, 2014)** and save the file

Unpack Siesta

Open a Terminal by clicking the Screen icon on your left; then, issue the following commands:

```
$ mkdir TUTORIAL
$ cd TUTORIAL
$ cp ../Downloads/siesta-trunk-462.tgz .
$ tar –zxvf siesta-trunk-462.tgz
$ cd siesta-trunk-462
$ ls –l
$ cd Docs
$ pdflatex siesta.tex
$ pdflatex siesta.tex
$ evince siesta.pdf &
```

Worth Reading: The Siesta Manual

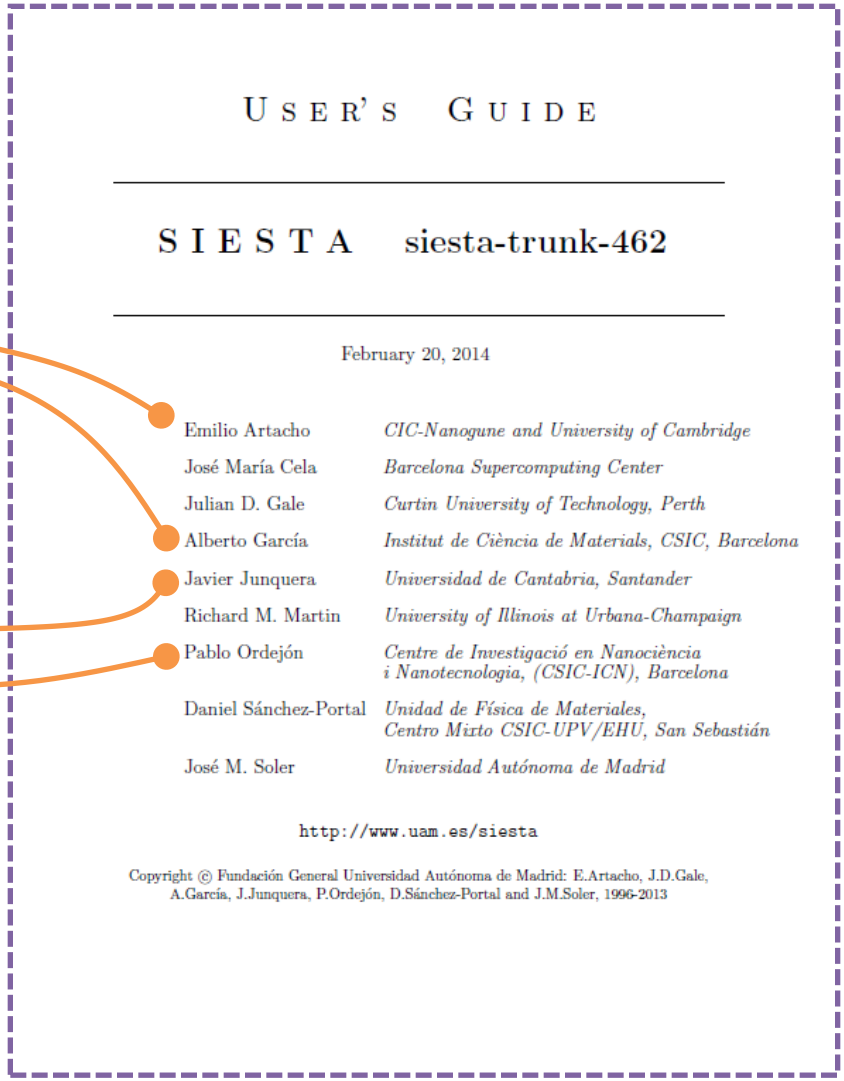
Developers (here today):




Contributors (here today):


Nick Papior Andersen


Oswaldo Dieguez



USER'S GUIDE

SIESTA siesta-trunk-462

February 20, 2014

Emilio Artacho	*CIC-Nanogune and University of Cambridge*
José María Cela	*Barcelona Supercomputing Center*
Julian D. Gale	*Curtin University of Technology, Perth*
Alberto García	*Institut de Ciència de Materials, CSIC, Barcelona*
Javier Junquera	*Universidad de Cantabria, Santander*
Richard M. Martin	*University of Illinois at Urbana-Champaign*
Pablo Ordejón	*Centre de Investigació en Nanociència i Nanotecnologia, (CSIC-ICN), Barcelona*
Daniel Sánchez-Portal	*Unidad de Física de Materiales, Centro Mixto CSIC-UPV/EHU, San Sebastián*
José M. Soler	*Universidad Autónoma de Madrid*

http://www.uam.es/siesta

What is Siesta?

SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms) is both a method and its computer program implementation, to perform electronic structure calculations and *ab initio* molecular dynamics simulations of molecules and solids. Its main characteristics are:

- It uses the standard Kohn-Sham selfconsistent density functional method in the local density (LDA-LSD) and generalized gradient (GGA) approximations, as well as in a non local functional that includes van der Waals interactions (VDW-DF).
- It uses norm-conserving pseudopotentials in their fully nonlocal (Kleinman-Bylander) form.
- It uses atomic orbitals as a basis set, allowing unlimited multiple-zeta and angular momenta, polarization and off-site orbitals. The radial shape of every orbital is numerical and any shape can be used and provided by the user, with the only condition that it has to be of finite support, i.e., it has to be strictly zero beyond a user-provided distance from the corresponding nucleus. Finite-support basis sets are the key for calculating the Hamiltonian and overlap matrices in $O(N)$ operations.
- Projects the electron wavefunctions and density onto a real-space grid in order to calculate the Hartree and exchange-correlation potentials and their matrix elements.
- Besides the standard Rayleigh-Ritz eigenstate method, it allows the use of localized linear combinations of the occupied orbitals (valence-bond or Wannier-like functions), making the computer time and memory scale linearly with the number of atoms. Simulations with several hundred atoms are feasible with modest workstations.
- It is written in Fortran 95 and memory is allocated dynamically.
- It may be compiled for serial or parallel execution (under MPI).

Starting from version 3.0, SIESTA includes the TRANSIESTA module. TRANSIESTA provides the ability to model open-boundary systems where ballistic electron transport is taking place. Using TRANSIESTA one can compute electronic transport properties, such as the zero bias conductance and the I-V characteristic, of a nanoscale system in contact with two electrodes at different electrochemical potentials. The method is based on using non equilibrium Green's functions

Lecture 2 (Artacho)

Lecture 6 (Junquera)

Lecture 7 (Artacho)

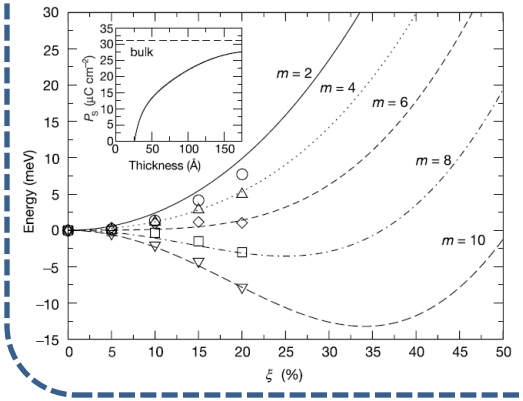
Lecture 10 (Ordejon)

Lecture 20 (Garcia)

Lectures 11—14 (Andersen, Ordejon)

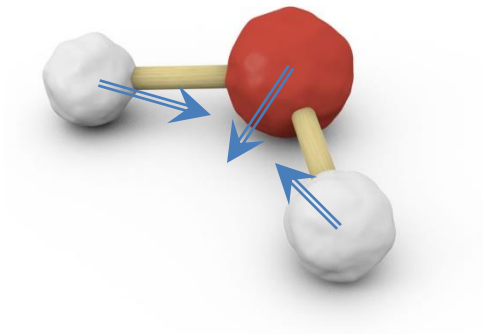
What Can Siesta Compute?

Energies



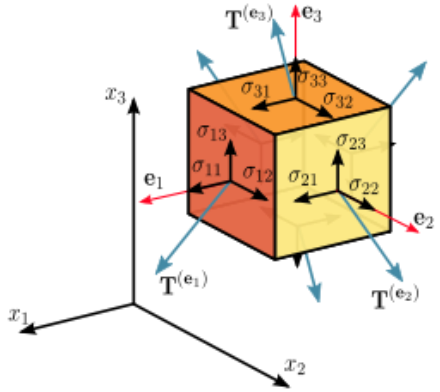
Junquera & Ghosez, Nature 2003

Forces



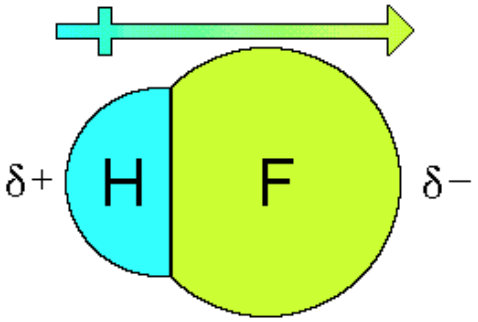
http://www.jameshedberg.com

Stress



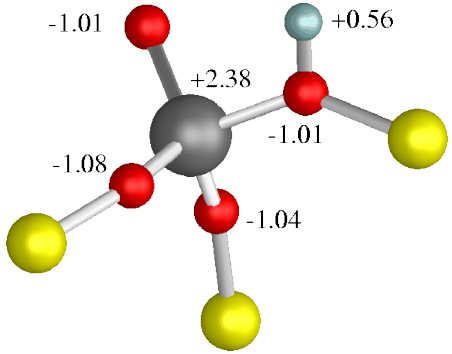
http://www.wikipedia.org

Electric Dipoles



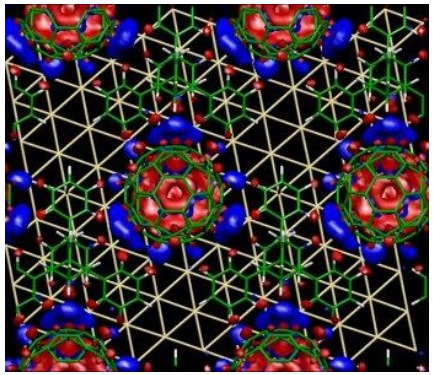
http://intro.chem.okstate.edu/

Charge Populations



http://www.tcm.phy.cam.ac.uk/

Electric Charge



http://departments.icmab.es/leem

What Can Siesta Compute?

And also (though not all options are compatible):

- Geometry relaxations, fixed or variable cell
- Constant-temperature molecular dynamics
- Variable cell dynamics
- Spin polarized calculations (collinear or not)
- k-sampling of the Brillouin zone
- Local and orbital-projected density of states
- COOP and COHP curves for chemical bonding analysis
- Dielectric polarization
- Vibrations (phonons)
- Band structures
- Ballistic electron transport (through TranSiesta)

Compiling Siesta

Issue the following commands:

```
$ cd ~/TUTORIAL/siesta-trunk-462
$ ls -l
drwxr-xr-x  2 dieguez  4096 Sep  3 15:36 Docs
drwxr-xr-x  9 dieguez  4096 Sep  3 15:33 Examples
drwxr-xr-x 10 dieguez 20480 Sep  3 17:33 Obj
drwxr-xr-x  3 dieguez  4096 Sep  3 15:33 Pseudo
-rw-r--r--  1 dieguez   682 Jul  9 12:22 README
-rw-r--r--  1 dieguez    95 Jul  9 12:22 README_TRANSIESTA
-rw-r--r--  1 dieguez   262 Jul  9 12:22 SIESTA_LICENCE
drwxr-xr-x 13 dieguez 12288 Sep  3 15:33 Src
drwxr-xr-x 99 dieguez  4096 Sep  3 15:33 Tests
drwxr-xr-x  4 dieguez  4096 Sep  3 15:33 Tutorials
drwxr-xr-x 38 dieguez  4096 Sep  3 15:33 Util
-rw-r--r--  1 dieguez    17 Jul  9 12:22 version.info
```

Source code:

- 269 Fortran-77 files
- 207 Fortran-90 files
- 15 shell scripts
- 23 m4 files
- 3 Matlab files
- 1 C file

185,263 lines of code

Compiling Siesta

```
$ cd Obj
$ sh ../Src/obj_setup.sh
$ ls ../Src/Sys
```

```
DOCUMENTED-TEMPLATE.make     hpcx.make                    marenostrum-mpi-32.make
README                       ibm-regatta-p4.make          marenostrum-mpi-64.make
altix-32b-par.make           ibmp3-mpi.make               marenostrum-mpi-essl-netcdf-64.make
altix-32b-ser.make           ibmp3.make                   marenostrum-mpi-metis-64.make
arch.make.curie              icmab-snake-mpi.make         marenostrum-mpi-netcdf-32.make
arch.make.fermi              ifort-netcdf.make            marenostrum-mpi-netcdf-64.make
arch.make.hermit             intel-checks.make            matgas-lam.make
arch.make.juqueen            intel-mkl.make               mn-32b-par.make
arch.make.marenostrum        intel-mpi-checks-metis.make  mn-32b-ser.make
arch.make.supermuc           intel-nolibs.make            mn-openmp.make
arina-mlib.make              intel10-openmpi.make         nano-intel-mpi-cdf.make
...                          ...                          ...
```

```
$ cp ../Src/Sys/gfortran.make arch.make
$ make
```

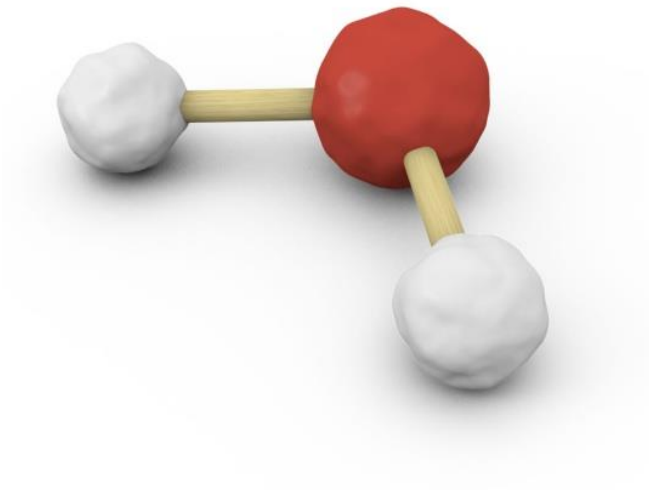
Compiling Siesta (2)

Siesta relies in external libraries, that need to be installed:

- Install LAPACK (very easy to do in Ubuntu):
  `$ sudo apt-get install liblapack-dev`
- Edit `arch.make`:
  Change the line…
  ```
  LIBS=
  ```
  into
  ```
  LIBS= -L /usr/lib –llapack –lblas
  ```
- Type **`make`**
- Type **`ls siesta`**

Our First Run: Main Input File

```
$ cd ~/TUTORIAL
$ mkdir h2o
$ cd h2o
$ cp ../siesta-trunk-462/Examples/H2O/h2o.fdf .
```



```
SystemName          Water molecule
SystemLabel         h2o
NumberOfAtoms       3
NumberOfSpecies     2

%block ChemicalSpeciesLabel
 1  8  O      # Species index, atomic
number, species label
 2  1  H
%endblock ChemicalSpeciesLabel

AtomicCoordinatesFormat  Ang
%block AtomicCoordinatesAndAtomicSpecies
 0.000  0.000  0.000  1
 0.757  0.586  0.000  2
-0.757  0.586  0.000  2
%endblock AtomicCoordinatesAndAtomicSpecies
```

Our First Run: Pseudopotential Files

```
$ cd ../siesta-trunk-462/Pseudo/atom
$ make
$ cd Tutorial/PS_Generation/O
$ cat O.tm2.inp
```

```
pg      Oxygen
        tm2  2.0
n=O  c=ca
      0.0       0.0       0.0       0.0
0.0       0.0
    1    4
    2    0     2.00      0.00
    2    1     4.00      0.00
    3    2     0.00      0.00
    4    3     0.00      0.00
  1.15     1.15     1.15     1.15
```

```
$ sh ../../Utils/pg.sh O.tm2.inp
$ cp O.tm2.psf ~/TUTORIAL/h2o/O.psf
$ cd ~/TUTORIAL/h2o
$ cp ../siesta-trunk-462/Examples/Vps/H.psf .
```

Our First Run: Executing Siesta

```
$ ln –s ../siesta-trunk-462/Obj/siesta .
$ ./siesta < h2o.fdf | tee h2o.out
```

Main output first lines…

```
Siesta Version: siesta-trunk-462
Architecture  : gfortran-nolibs
Compiler flags: gfortran -O2
PP flags      : -DGFORTRAN -DFC_HAVE_FLUSH -DFC_HAVE_ABORT
SERIAL version

* Running in serial mode
>> Start of run:   6-SEP-2014  15:23:36

                           ***********************
                           *  WELCOME TO SIESTA  *
                           ***********************

reinit: Reading from standard input
************************** Dump of input data file ****************************
```

Our First Run: Executing Siesta

Energy…

```
siesta: Final energy (eV):
siesta:  Band Struct. =    -104.303601
siesta:       Kinetic =     353.764173
siesta:       Hartree =     385.167485
siesta:    Ext. field =       0.000000
siesta:   Exch.-corr. =    -113.205313
siesta:  Ion-electron =   -1080.072411
siesta:       Ion-ion =     -11.477707
siesta:       Ekinion =       0.000000
siesta:         Total =    -465.823773
```

Forces…

```
siesta: Atomic forces (eV/Ang):
siesta:      1    0.000000   -0.760230    0.000000
siesta:      2    0.700706    0.352159   -0.000000
siesta:      3   -0.700706    0.352159   -0.000000
siesta: ----------------------------------------
siesta:    Tot    0.000000   -0.055912   -0.000000
```

Geometry Optimization

1. Make a new directory, **`~/TUTORIAL/h2o_opt`**
2. Copy the same main input file and the pseudopotential files from the previous run to this new directory
3. Add to **`h2o.fdf`** the following lines:

```
MD.TypeOfRun        CG
MD.NumCGSteps       30
MD.MaxForceTol      0.02 eV/Ang
WriteCoorXmol       .true.
```

4. Run Siesta using this new input files

Geometry Optimization: ANALYSIS

1. **What are the forces on the atoms now?**
2. **Plot the total energy versus the number of steps in the optimization process**
   Create a file name `energy.dat` with two colums, the first with the step number and the second with the total energy; then type in your terminal the command `xmgrace energy.dat &`
3. **Visualize the atoms in the molecule, and measure the bond lengths and the bond angle; compare your results with online experimental data**
   Type in your terminal `jmol h2o.xyz &`; then display a menu by right-clicking in the background, and choose “Measurements” and the right options for lengths and angles
4. **Find out the magnitude of the electric dipole, and compare it with online experimental information**

Geometry Optimization: ANALYSIS

1. **What are the forces on the atoms now?**

BEFORE…

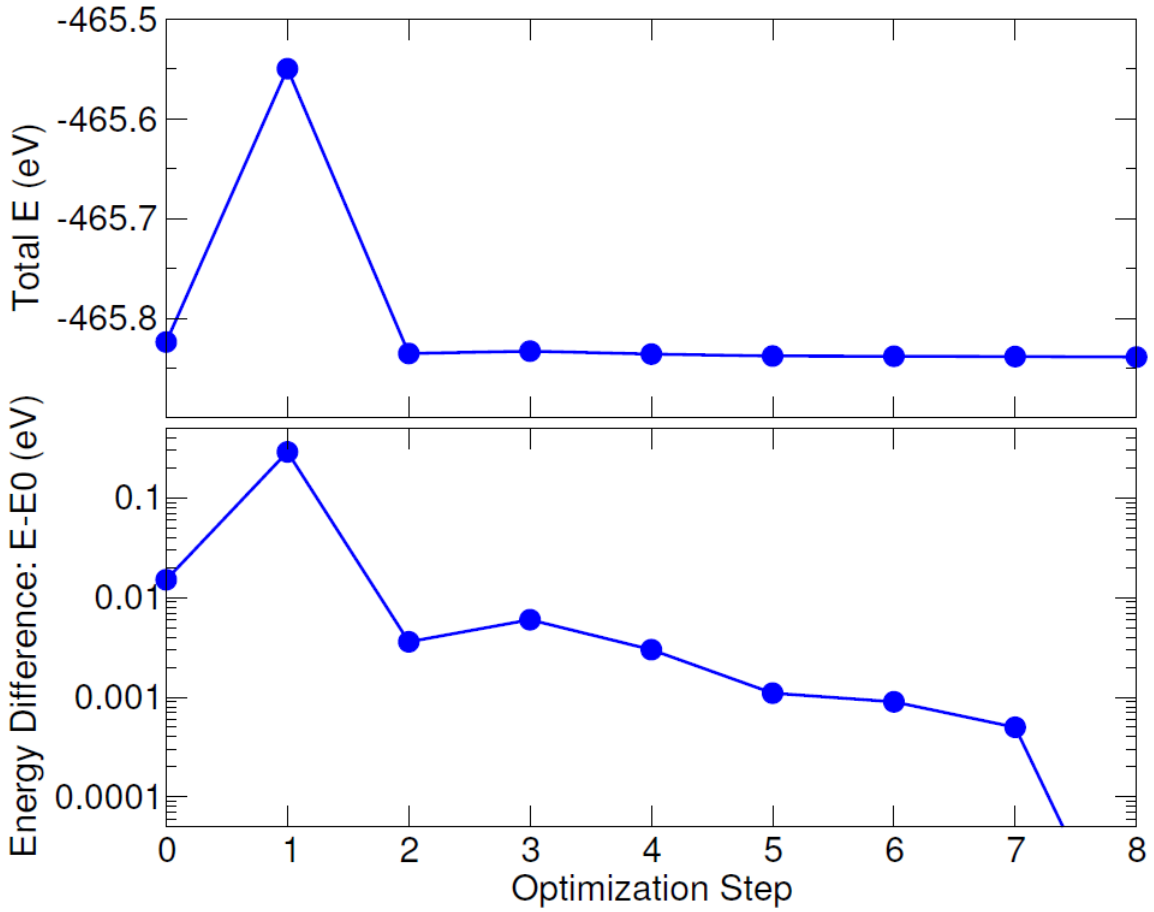
```
siesta: Atomic forces (eV/Ang):
siesta:      1    0.000000   -0.760230    0.000000
siesta:      2    0.700706    0.352159   -0.000000
siesta:      3   -0.700706    0.352159   -0.000000
siesta: ----------------------------------------
siesta:    Tot    0.000000   -0.055912   -0.000000
```

NOW (in file h2o.FA) …

```
     1    0.000000   -0.014940    0.000000
     2   -0.013803   -0.015041   -0.000000
     3    0.013803   -0.015041   -0.000000
```

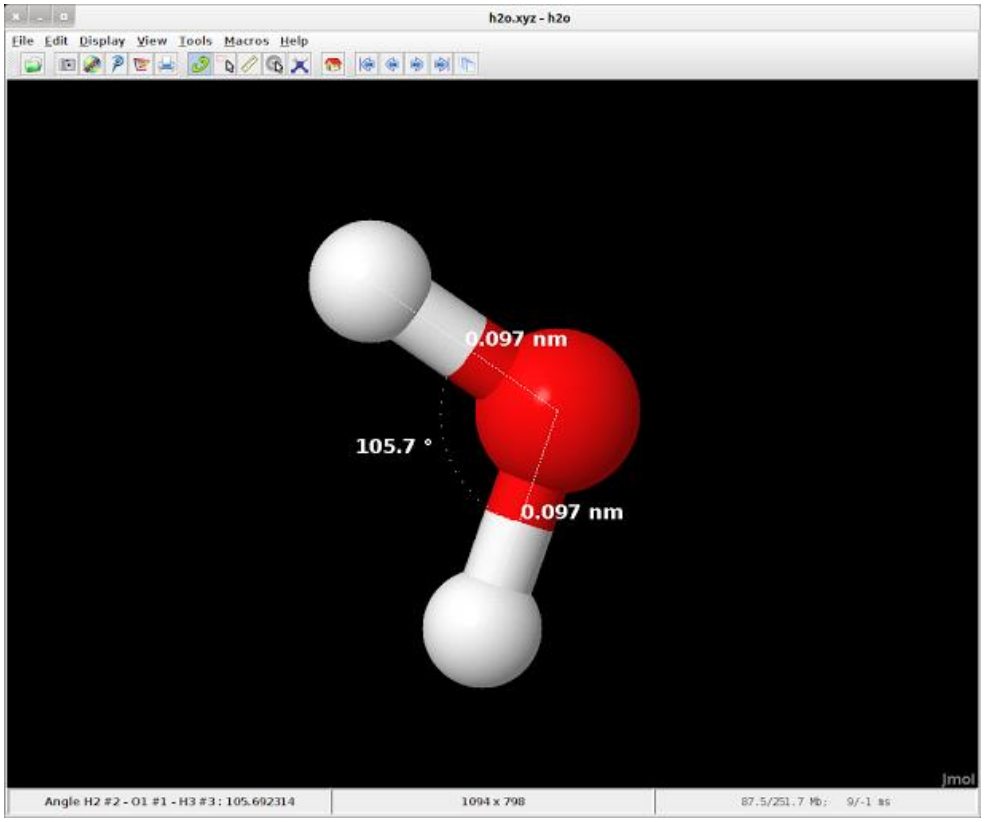
Geometry Optimization: ANALYSIS

2. **Plot the total energy versus the number of steps in the optimization process**



Geometry Optimization: ANALYSIS

3. **Visualize the atoms in the molecule, and measure the bond lengths and the bond angle; compare your results with online experimental data**



EXPERIMENTAL
bond: 0.96 Å
angle: 104.5°
(wikipedia.org)

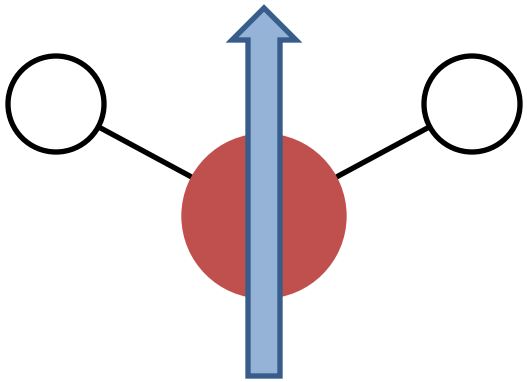
Geometry Optimization: ANALYSIS

4. **Find out the magnitude of the electric dipole, and compare it with online experimental information**

RESULT (in file h2o.out) …

```
siesta: Electric dipole (a.u.)  =    -0.000000    0.555056    0.000000
siesta: Electric dipole (Debye) =    -0.000000    1.410812    0.000000
```

```
outcoor: Relaxed atomic coordinates (Ang):
   -0.00000000   -0.00037891   -0.00000000   1       1  O
    0.77464263    0.58662409    0.00000000   2       2  H
   -0.77464263    0.58662409    0.00000000   2       3  H
```



EXPERIMENTAL

dipole: 1.85 D

(wikipedia.org)

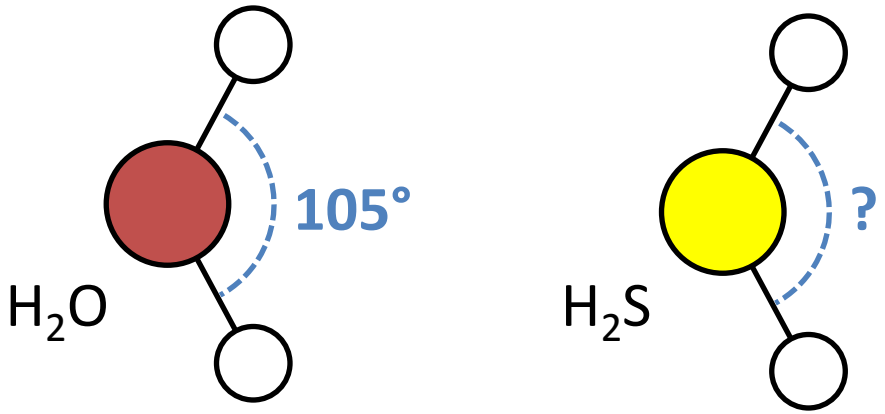
Understanding VS Numbers

Siesta can provide you with numerical answers (a good thing)…

BUT

…computation is no substitute for understanding

EXERCISE:
Is the bond angle in H_2S smaller or larger than in H_2O?



Mon, 8 Sep 2014

08:45 -- REGISTRATION and refreshments
09:30 -- Introduction to the role of first-principles calculations
10:15 -- DFT and general notions about first-principles codes
11:00 -- Coffee break
11:30 -- Overview of Siesta
13:00 -- Lunch
14:00 -- Examples of practical calculations with SIESTA
15:30 -- Coffee break
16:00 -- Examples of practical calculations with SIESTA II
17:30 -- (End of day 1 lectures)

LUNCH
Gilman Building
(Humanities)