

EXERCISE 1

BASIC SIESTA EXECUTION

In this exercise we will focus on the basic input parameters and outputs of the SIESTA program. We will study two simple molecules (CH_4 and CH_3)

Enter in the directory *CH4*. There, you will find an input file for the SIESTA program named *ch4.fdf* along with the files *C.psf* and *H.psf* containing the information about the pseudopotentials. The *ch4.fdf* file sets the value of several parameters which specify both the system we want to study and the accuracy of the calculation. Of course the parameters specifying the system are mandatory (number of atoms, atomic numbers, label associated with each atomic specie, etc..). All other parameters have some default values and, in principle, it is not necessary to explicitly include them in the input file. However, it is important to note that the default values do not always guarantee a converged calculation.

File *ch4.fdf*

Edit the file. You can find first the inputs that specify the system. Pay special attention to the block *ChemicalSpeciesLabel*. In this block you assign an index and a label to each atomic species. The label will allow to recognize the files containing the information about the pseudopotential and the basis set (when provided).

Check the input of the coordinates (they are just some guess coordinates, not the optimized equilibrium ones).

The file *ch4.fdf* contains the most important parameters to take into account to perform a molecular calculation. Namely:

- i) Those defining the size and localization of the basis set.
- ii) The parameter *MeshCutoff*, controlling the fineness of the real-space grid used to compute the integrals for the matrix elements of the Hamiltonian.
- iii) Those that control the self-consistent cycle.

Automatic basis set

The number of orbitals per atom is defined by the parameter *PAO.BasisSize*. In this case we have fixed this value to a minimal basis (SZ) for quick, cheap calculation, looking for qualitative results, rather than quantitative results. The SIESTA default, however, is DZP, corresponding to fairly

Run the program
siesta < ch4.fdf > ch4.out

Edit *ch4.out* once the program has finished. Explore carefully the output of SIESTA for this calculation. The file contains first a copy of the input. After this, the atomic information is displayed. Details about the read pseudopotential and the basis set generated are given for each species. Observe what the program is telling you about the number of non-local pseudopotentials that are being used, and how this can be changed. Of special interest is the block named *PAO.Basis* appearing at the end of the atomic information. This gives at the same time a summary of all the parameters used to generate the automatic basis set and an alternative way to provide the input of the basis sets. This input format allows the user to modify each parameter used in the basis set generation at will. After the atomic information a summary of the values used in the calculation for the most important parameters is given.

Convergence of the SCF cycle

Before considering the accuracy of the calculations with respect to the basis set or the fineness of the real-space mesh it is crucial to be able to converge the calculation reaching the self-consistent condition.

Two things are crucial here. First, to be able to decide whether selfconsistency has been reached, and second, to be able to control how the electronic density obtained in a given step is mixed with those obtained in previous steps.

The SCF condition is monitored in SIESTA by looking at the maximum difference between the matrix elements of the new and old density matrix. The tolerance for this change is set by *DM.Tolerance*. The default is 10^{-4} , which is a rather good value. Usually you should not change this value.

The mixing between old and new density matrix is control by *DM.MixingWeight* for the simple linear mixing:

$$\rho_{n+1}^{in} = \alpha \rho_n^{out} + (1 - \alpha) \rho_n^{in}$$

Play with this parameter to see if you can accelerate the convergence.

You have probably noticed that using large values (close to 1), reaching convergence becomes extremely difficult or even impossible. However, if you use a large value, but now set the parameter *DM.NumberPulay* to an integer value large than 1, you will see that the SCF convergence is reached in a few iterations! You are now using the Pulay mixing scheme, in which the input for the next step is obtained by mixing several previous steps (defined by *DM.NumberPulay*), instead of only the last one. You can check the manual for the details. Play with all the parameters to check if you can find optimum values for a fast convergence.

Meshcutoff

We can now play a bit with the fineness of the real-space grid. Use a unreasonable low value for the the parameter *MeshCutoff* (may be 10-30 Ry) and check how total energy and forces (you can

find the forces on file *ch4.FA*). Try to determine the minimum value of the *MeshCutoff* parameter that gives an energy converged to 0.1 eV.

Unit cell

SIESTA uses periodic boundary conditions. This means that we always do calculations for an infinite collection of regularly spaced molecules. However if we want to simulate an isolated molecule it is important to have enough distance between the molecule and its neighboring images. Up to now we have been using an unit-cell automatically generated by the program, and now we will define it explicitly to check the effect of changing the cell size. Enter the directory *cubic*. The file *ch4_cubic.fdf* specifies the unit cell used for the calculation. Is a cubic cell with a 15 Ang lattice parameter. Check the format used to define the cell and the lattice parameter. Notice that, once the unit cell is defined, the atomic coordinates can be defined in relative units (check the manual). You can play with the size of the lattice parameters to go from interacting molecules to effectively isolated ones. Look at the variation in the total energy as a function of the cell size, to see how the interaction between molecules decreases with increasing distance between images.

DFT functional

Up to now we have been using LDA for our calculations. However, it is also possible to use GGA. Enter the directory *gga*. You can find the file *ch4_gga.fdf*. Using this input file the program will use the Perdew-Burke-Ernzerhof GGA functional. Run the program and look for possible *WARNING* or *ERROR*. In fact, you can see that there is a warning. The code does not like that you are using GGA with a pseudopotential generated using LDA, this is not consistent!. Fortunately, we have produced also the pseudopotential using GGA for you. They are in the files *C.gga.psf* and *H.gga.psf*. Modify the input file so you will use these files and check whether the warning disappears from the output.

Structural optimization

Enter the directory *relax*. Using the file *ch4_relax.fdf* we can perform now a structural optimization using the conjugate gradient algorithm. Check the manual to understand the meaning of the lines added to the input. Relax the structure for several basis set sizes (SZ, DZ, DZP) and check the differences on geometry and total energy. The file *ch4.ANI* contains all the structures generated during the relaxation in XYZ format appropriate for several programs, like Xmol or Molekel.

Spin polarization

Enter in the directory *CH3*. Now we are going to perform calculations for the molecule CH₃. Run the program with the input file *ch3.fdf*. You will obtain the LDA optimized geometry of the molecule. However, it is interesting to notice that this molecule contains an unpaired electron. Therefore the system should show some spin polarization. The file *ch3_spin.fdf* contains the input to perform a spin polarized calculation, more appropriate for this system. Compare the results of this calculations with those of the previous one. An interesting point has to be stressed here. To obtain spin polarization we need to break the symmetry between spin up and down. Therefore, if the symmetry is somehow imposed or assumed in the initial configuration the results will never be spin-polarized. You can check this using the file *ch3_initspinzero.fdf* (check the manual to see the meaning of the block *InitSpin* used in this input file)

Plotting densities

Enter *3Dplot*. Check the input file *ch3_3Dplot.fdf*. Using this file, SIESTA will produce a file containing the values of the self-consistent electronic density on the real-space mesh (file called *ch3.RHO*) and the those of the density associated to the HOMO of the molecule (file called *ch3.LDOS*). The files can be translated to a format that can be visualized with the program Molekel using the utility program *grid2cube.f*. For this, after running the program you need to execute *grid2cube.x < grid2cube.dat* (Edit the file *grid2cube.f* to get detail information about this program and its input)

Visualize the isosurfaces of electronic density ($\rho(x,y,z)=\text{const}$) for several values of the density (i.e. const) and each spin. Do the same for the density associated with the molecular HOMO. Modify the block LocalDensityOfStates to plot the density associated with different molecular orbitals lying in different energy windows.