

SIESTA Tutorials



SIMUNE Atomistics S.L.

Tolosa Hiribidea, 76
20018, San Sebastian, Spain

SIESTA tutorials for personal use only. The material cannot be distributed to third parties



www.simune.eu

Unauthorized copying or posting is an infringement of copyright

Contents

Contents	i
Introduction	1
Obtaining a well converged SIESTA calculation	2
Tutorial Overview	3
Tutorial Remarks	4
1 The SIESTA Package: Code Compilation	5
1.1 Code compilation. Minimal instructions	6
1.2 Further notes	9
2 SIESTA execution	11
2.1 Requirements	11
2.2 How to run SIESTA in serial mode	11
2.3 Output: the header	12
3 Basic Example: energy optimization of H2 molecule	13
3.1 SIESTA Input: The FDF file	13
3.1.1 Characteristics of the FDF	13
3.1.2 General descriptors - Structural variables	14
0.1 Electronic Density Convergence & Structural Optimisation	1
0.1.1 Tutorial Material	1
0.1.2 Tutorial Instructions	1
4 Atomic orbital of finite range as a basis set: basis-set convergence	26
4.1 Scientific Background: Atomic orbital of finite range as a basis set	26
4.2 Tutorial Material	28
4.3 Tutorial Instructions	28
4.4 Relevant references	33
5 Optimization of material properties: Equilibrium bulk structure	35
5.1 Tutorial Material	35
5.2 Tutorial Instructions	36

6 Optimization of material properties: Band Structure	40
6.1 Tutorial Material	40
6.2 Tutorial Instructions	41
7 Optimization of material properties: Density Of States (DOS)	45
7.1 Scientific Background: Density of States (DOS), and Projected Density of States (PDOS) .	45
7.2 Tutorial Material	46
7.3 Tutorial Instructions	46
8 Optimization of material properties: Charge Density	51
8.1 Tutorial Material	51
8.2 Tutorial Instructions	52
9 Molecular Dynamics in SIESTA	56
9.1 Introduction	56
9.2 Scientific Background of MD	56
9.3 When Do We Use MD?	58
9.4 Tutorial Material	58
9.5 Tutorial Instructions	58
9.6 Input file .fdf	59
9.6.1 Standard MD SIESTA Calculation	59
9.6.2 Output of a MD in SIESTA	59
9.6.3 Plot MDE	60
9.6.4 MD SIESTA Calculation: Defying the Time Step	61
9.6.5 MD SIESTA Calculations for a Range of Time Steps	62
9.7 Visualization	63
10 LDA+U Approximation in SIESTA	64
10.1 Scientific background: LDA+U Approximation	64
10.2 Tutorial Material	65
10.3 Tutorial Instructions	65
10.3.1 Executing SIESTA: FeO - GGA	66
10.3.2 LDA+U: Input Parameters	68
10.3.3 Convergency Issue	70
10.3.4 Executing SIESTA: FeO - LDA+U	70
10.4 Plotting the LDA+U Projectors	72

3.2 Electronic Density Convergence & Structural Optimisation

In this section of the tutorial the topics of electronic density convergence, and structural optimisation are addressed. A brief description of the Self-Consistent Field method (SCF) used to achieve **self-consistent electron density** is provided. The tutorial focuses on conjugate gradient method (CG) for **molecular geometry optimisation**.

3.2.1 Tutorial Material

The T3 folder contains the files that SIESTA requires to optimize the energy of a H₂ molecule,

1. **H2.fdf**: SIESTA input file
2. **H.psf**: Pseudopotential file
3. **get-SCF.sh**: Bash script to obtain and store information on the electronic convergence for each conjugate gradient cycle
4. **plot-SCF.sh**: Bash script to plot the data extracted by **get-SCF.sh**

3.2.2 Tutorial Instructions

1. Explore the input of SIESTA

Inspect the file, **H2.fdf**.

[more H2.fdf](#)

H2.fdf contains parameters specifying the system we want to study (these parameters were described in the first section of this tutorial), but also parameters specifying the accuracy of the calculation (Self Consistent options and Conjugate Gradient Method parameters).

```
# Self Consistent Options
# =====
MaxSCFIterations      200
DM.MixingWeight       0.25
DM.NumberPulay        0
DM.Tolerance          1.d-5

# Conjugate Gradient Method
# =====
MD.TypeOfRun          CG
MD.NumCGsteps         50
```

MD. MaxCGDispl	0.4 Ang
MD. MaxForceTol	0.04 eV/Ang

Electronic structure - Self-consistency

SIESTA makes use of the fully self-consistent Kohn-Sham functional to calculate energies and forces. This method consists in defining an initial electron density ($\rho_{in}^n(\mathbf{r})$) to solve Kohn-Sham equations. Then, from the solution of these equations a new electronic density ($\rho_{out}^n(\mathbf{r})$) is calculated. The cycle is repeated until a self-consistent electron density is achieved, see Figure 1.

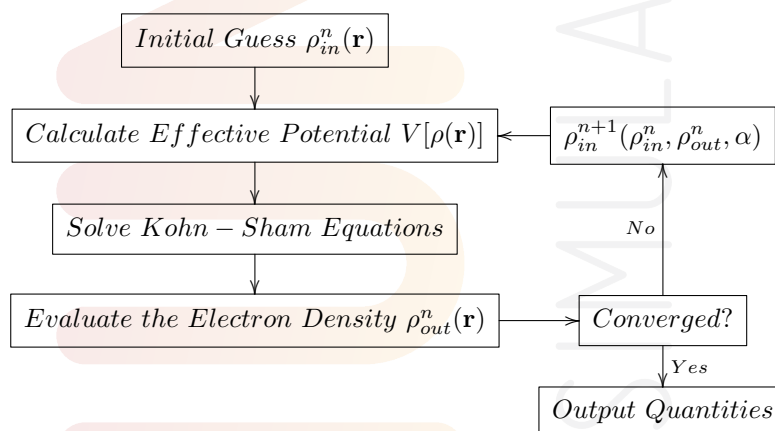


Figure 3.2: Schematic representation of the Self Consistent Field Method (SCF)

Two things are crucial here. First, to be able to decide whether self-consistency 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. In this tutorial are used the following parameters to describe the self-consistency cycles,

(a) **Number of cycles**

The parameter *MaxSCFIterations* refers to the maximum number of Self-consistent iterations.

MaxSCFIterations	200	#default value: 50 cycles
------------------	-----	---------------------------

(b) **Mixing**

Proportion α of output Density Matrix ($\alpha\rho_{out}^n$) to be used for the input Density Matrix of next SCF cycle (ρ_{in}^{n+1}) (linear mixing). The mixing between old and new density matrix (α) is controlled by *DM.MixingWeight*.

$$\rho_{in}^{n+1} = \alpha\rho_{out}^n + (1 - \alpha)\rho_{in}^n \quad (3.1)$$

DM. MixingWeight	0.25	#default value: 0.25
------------------	------	----------------------

Pulay mixing is the method of choice for accelerating the convergence of the SCF cycle. The use of this method generally accelerates convergence quite significantly, and can reach convergence in cases where linear mixing cannot.

DM.NumberPulay controls the Pulay convergence accelerator.

DM. NumberPulay	1	#default value : 0
-----------------	---	--------------------

If *DM.NumberPulay* is 0 or 1, simple linear mixing is performed. If *DM.NumberPulay* is different than 0 or 1, the guess for the $n + 1$ iteration is constructed using the input and output matrices of the *DM.NumberPulay* former SCF cycles, in the following way,

$$\bar{\rho}_{in}^{n+1} = \alpha p \bar{\rho}_{out}^n + (1 - \alpha p) \bar{\rho}_{in}^n \quad (3.2)$$

where $\bar{\rho}_{out}^n$ and $\bar{\rho}_{in}^n$ are constructed from the previous $N = \text{DM.NumberPulay}$ cycles:

$$\bar{\rho}_{out}^n = \sum_{i=1}^N \beta_i \rho_{out}^{(n-N+i)} ; \quad \bar{\rho}_{in}^n = \sum_{i=1}^N \beta_i \rho_{in}^{(n-N+i)} \quad (3.3)$$

The values of β_i are obtained by minimizing the distance between the output and input density matrices. The value of αp is given by default by variable *DM.MixingWeight*.

(c) **Density Convergence Criterion**

The self-consistency has been achieved when the maximum difference between the output and the input on each element of the Density Matrix in a SCF cycle is smaller than the specified value for *DM.Tolerance*.

DM. Tolerance	1.d-5	#default value 1.d-5
---------------	-------	----------------------



Gemetry optimization options: Conjugate Gradient (CG) Method

Structural relaxation is not specific of siesta method, but is implemented to provide a more complete simulation package.

Several options for Molecular Dynamics (MD) and structural optimizations are implemented in SIESTA (*MD.TypeOfRun*). In this tutorial, we focus on the description of structural optimization by conjugate gradient method (CG).

MD.TypeOfRun	CG
--------------	----

CG Method has an outer geometry loop: it computes the electronic structure (and thus computes the forces and stresses) for a given geometry, updates the atomic positions accordingly (and maybe the cell vectors) and moves on to the next cycle. See Figure 2.

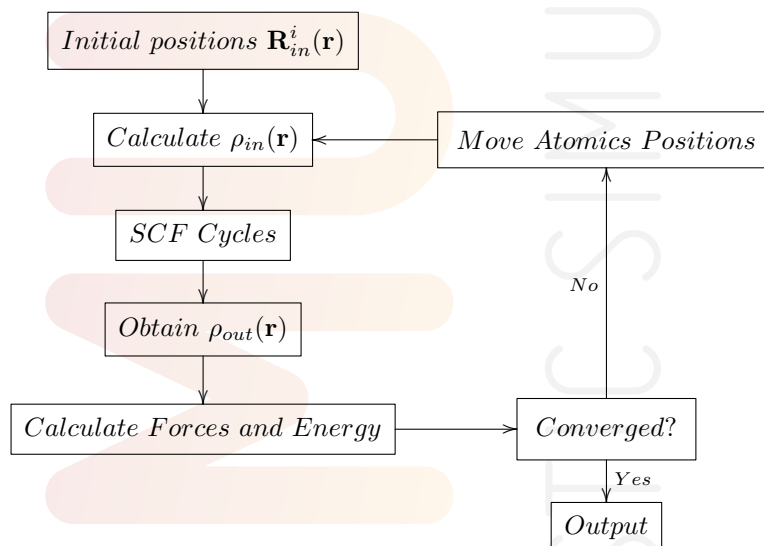


Figure 3.3: Schematic representation of the Conjugate Gradient (CG) Method

The parameters used in this tutorial to control the CG optimization are,

(a) **Number of cycles**

The maximum number of conjugate gradient minimization moves is specified by *MD.NumCGsteps*. The minimization stops if tolerance is reached before (see *MD.MaxForceTol*).

MD.NumCGsteps	50	# default value: 0
---------------	----	--------------------

(b) **Constrain**

The maximum atomic displacements in an optimization move is defined by *MD.MaxCGDisp*.

MD.MaxCGDisp	0.4 Ang	#default value: 0.2 Bohr
--------------	---------	--------------------------

(c) **Convergence Criteria**

The force tolerance in coordinate optimization is also defined. Run stops if the maximum atomic force is smaller than *MD.MaxForceTol*.

MD. MaxForceTol	0.04 eV/Ang	#default value: 0.04 eV/Ang
-----------------	-------------	-----------------------------

2. **SIESTA execution**

Type in the console the following command,

```
SIESTA<H2.fdf>H2.txt
```

where *H2.fdf* and *H2.txt* are the input and output files, respectively. And

```
SIESTA
```

must point to the *siesta* executable.

3. **Output inspection**

Explore carefully the output of SIESTA for this calculation.

```
more H2.txt
```

Following the output header, with information described in T2, *H2.txt* contains a copy of the input,

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

reinit: Reading from standard input
***** Dump of input data file *****
# Name and Label
___
___
___
___
***** End of input data file *****
```

After this information, details about the read pseudopotentials and basis set are given for each of the species. *H2.txt* provides also information on the electronic density convergence of the H₂ molecule in each of the SCF cycles, and on structural optimisation,

Self consistent field cycles

The fully self-consistent Kohn-Sham method, described in Figure 1 is used to calculate energies and forces on the H₂ molecule.


```
cat H2.txt|grep scf:
```

scf:	iscf	Eharris (eV)	E_KS(eV)	FreeEng (eV)	dDmax	Ef (eV)
scf:	1	-30.2258	-25.6734	-25.6734	0.79750	-2.8542
scf:	2	-30.0692	-30.0298	-30.0298	0.07486	-1.1147
.
scf:	26	-30.0627	-30.0626	-30.0626	0.00001	-1.4183
scf:	27	-30.0627	-30.0626	-30.0626	0.00001	-1.4183
scf:	iscf	Eharris (eV)	E_KS(eV)	FreeEng (eV)	dDmax	Ef (eV)
scf:	1	-25.7590	-24.5530	-24.5530	0.33031	-4.0395
scf:	2	-25.5649	-25.5316	-25.5316	0.07358	-1.7252
.

Conjugate gradient cycles

Geometry optimization is performed using the Conjugate Gradient (CG) method. The following command line can be used to check the convergence,

```
cat H2.txt |grep -A5 "Atomic forces (eV/Ang):"|grep Max
```

Max	3.690665
Max	44.173741
.	.
Max	0.072010
Max	0.001324

By default, the geometry optimization stops when the maximum force is lower than 0.04 eV/\AA (*MD.MaxForceTol*).

Atomic positions

The optimized molecular geometry can be inspected after electronic density convergence and structural optimization. The optimized molecular structure is stored in **H.STRUCT.OUT**. Compare the optimized atomic positions with those initial atomic positions that were specified in the input **H2.fdf**.

```
cat H2.fdf |grep -A12 "LatticeConstant"
```

LatticeConstant	10.00	Ang
%block	LatticeVectors	
1.0	0.0	0.0
0.0	1.0	0.0
0.0	0.0	1.0
%endblock	LatticeVectors	

```
AtomicCoordinatesFormat Ang
%block AtomicCoordinatesAndAtomicSpecies
  4.50000    5.000000    5.000000    1
  5.50000    5.000000    5.000000    1
%endblock AtomicCoordinatesAndAtomicSpecies
```

```
cat H2.STRUCT_OUT
```

```

10.000000000    0.000000000    0.000000000
 0.000000000    10.000000000    0.000000000
 0.000000000    0.000000000    10.000000000
 2
1 1 0.460375604 0.500000000 0.500000000
1 1 0.539624396 0.500000000 0.500000000
```

What is the difference in the atomic positions before and after geometry optimization? Is there any difference in the lattice parameter?

4. Analysis of Electronic Density convergence

We have created for your convenience bash scripts to collect data from each CG cycle ([get-SCF.sh](#)) method and to perform a graphic analysis of the self-consistent field method (SCF) convergence for each of the CG cycles ([plot-SCF.sh](#)).

Type first,

```
sh get-SCF.sh H2.txt
```

to create files storing information on each CG cycle ([CG-Cycles-1](#), [CG-Cycles-2](#), etc).

You can use now the script [plot-SCF.sh](#) to perform a graphic analysis of the SCF convergence on each of the CG steps.

```
gnuplot -e "File='CG-Cycle-2.txt'" plot-SCF.gplot
```

A figure as the one shown in [Figure 3](#) is created ([convergence-SCF.png](#)).

Remember that the accuracy of the calculation of energies and forces in a SCF loop can be tuned by specifying the number of maximum and minimum number of SCF iterations per time step, mixing options, etc. Play with [DM.MixingWeight](#) to see if you can accelerate the convergence.

You will 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

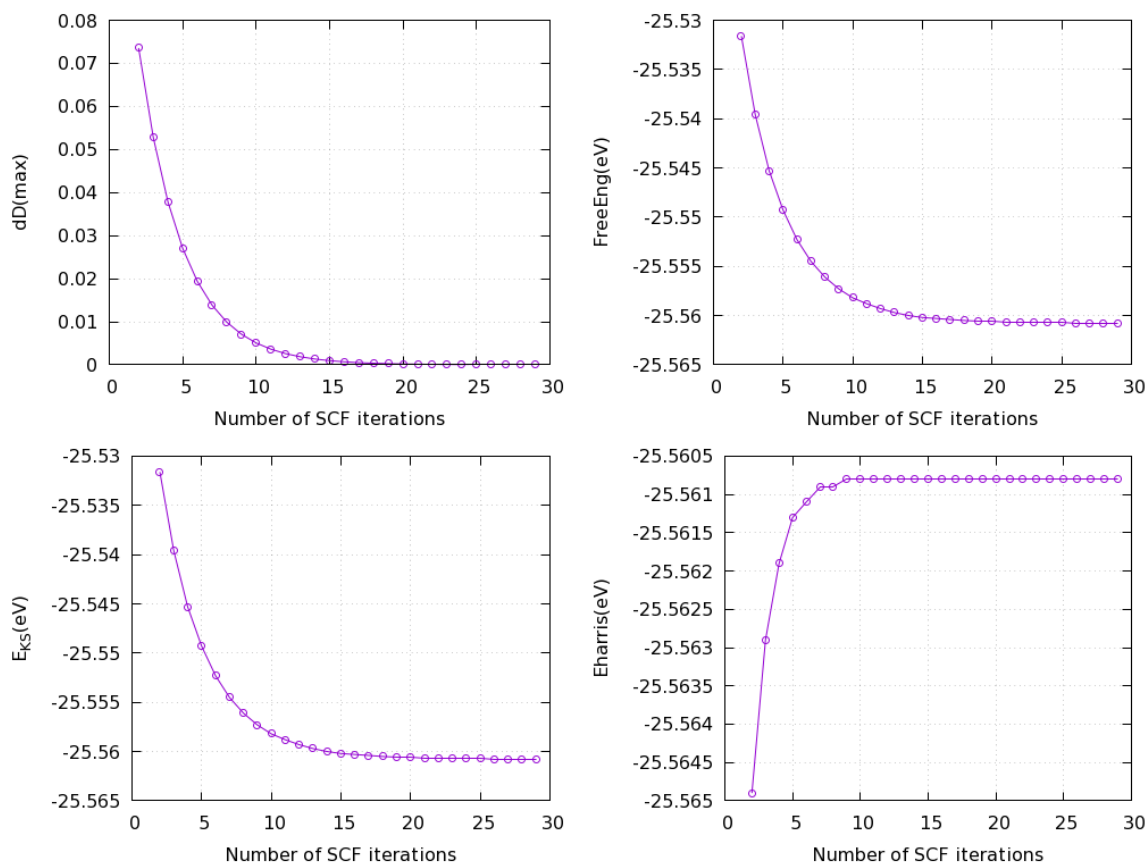
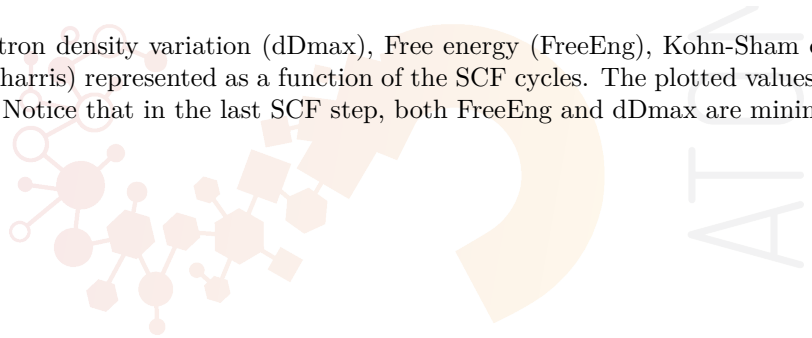


Figure 3.4: Electron density variation (dDmax), Free energy (FreeEng), Kohn-Sham energy (E_{KS}), and Harris energy (Eharris) represented as a function of the SCF cycles. The plotted values correspond to CG cycle number 2. Notice that in the last SCF step, both FreeEng and dDmax are minimal.



step is obtained by mixing several previous steps (defined by *DM.NumberPulay*), instead of only the last one.

Play with all the parameters to check if you can find optimum values for a fast convergence.

