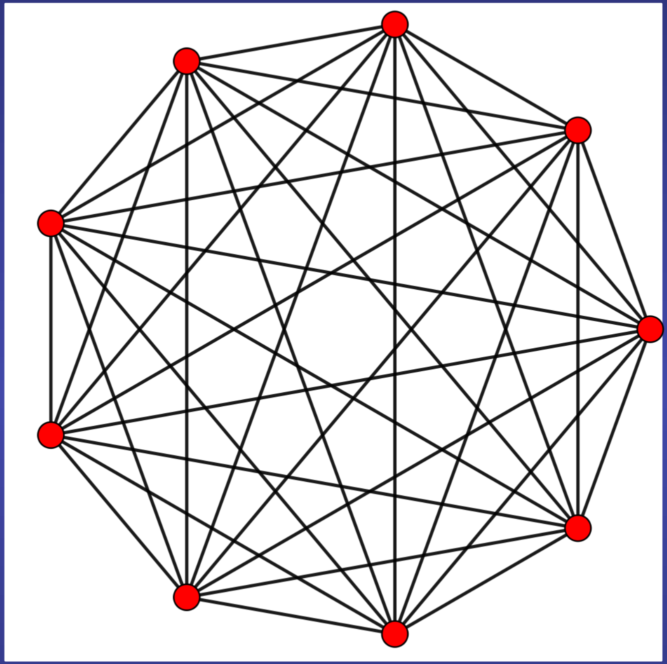
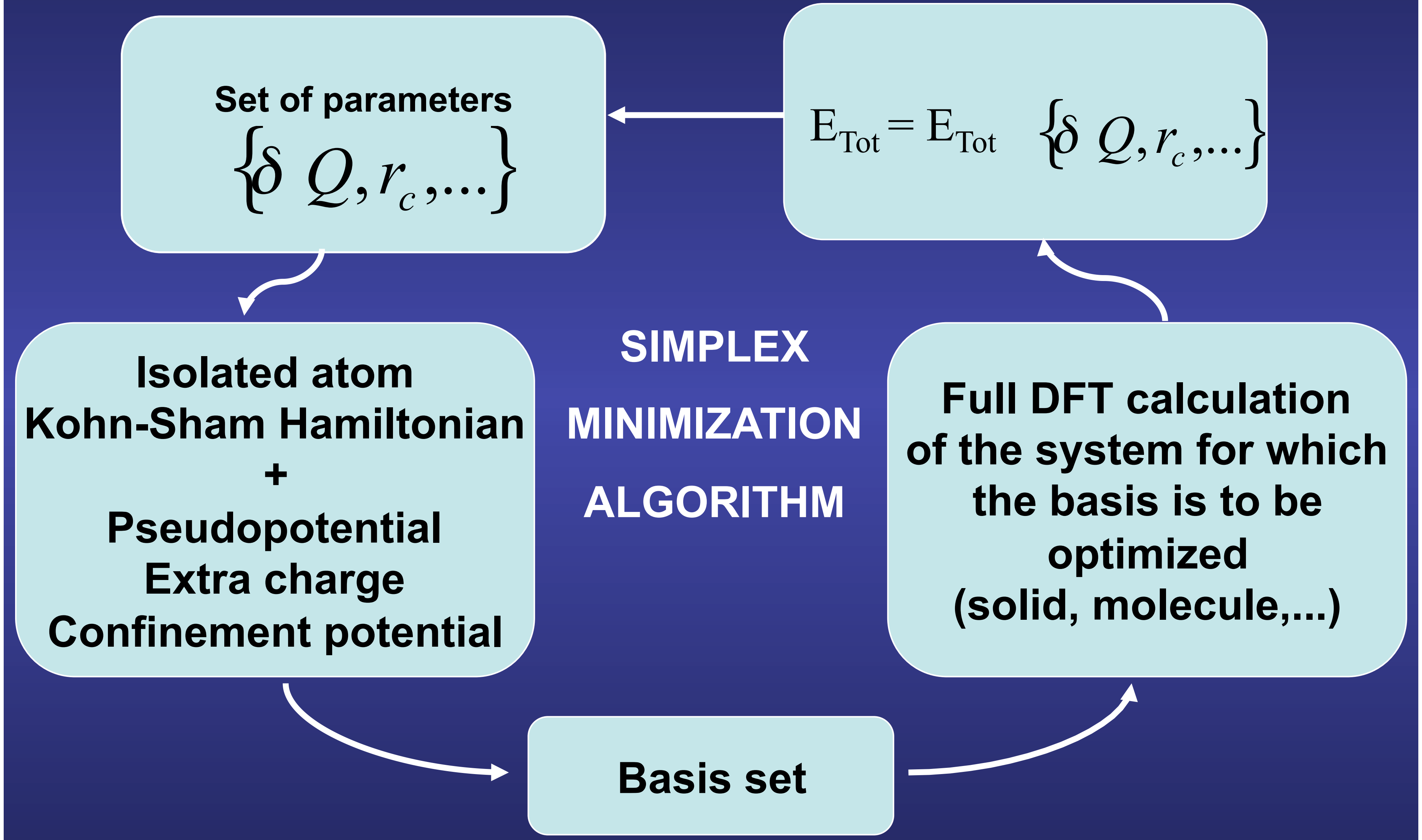
How to optimize automatically the parameters that determine the quality of the basis



Alberto García

Javier Junquera

Optimization of the parameters that define the basis set: the Simplex code


Set of parameters
$\{\delta Q, r_c, \ldots\}$
$E_{Tot} = E_{Tot} \{\delta Q, r_c, \ldots\}$
Isolated atom
Kohn-Sham Hamiltonian
+
Pseudopotential
Extra charge
Confinement potential
SIMPLEX
MINIMIZATION
ALGORITHM
Full DFT calculation of the system for which the basis is to be optimized (solid, molecule,...)
Basis set

J. Junquera et al. Phys. Rev. B 64, 235111 (2001)

Compiling the Simplex code to automatically optimize the basis set

Assuming that you have downloaded and unpacked the Siesta distribution in a directory ~/siesta, go to the ~/siesta/Util/Optimizer directory

```
$cd ~/siesta/Util/Optimizer
```

and type

```
$ make
```

This will compile the simplex using the same arch.make file as the one you to compile siesta

Compiling the Simplex code to automatically optimize the basis set

Create a directory to run the simplex for the desired system (in this particular case, the water molecule), and copy there

The pseudopotential files

The Siesta input file, that in this case will be called **TEMPLATE**

A file with the list of variables over which to optimize, and their ranges of variability, called **VARS**

A file with the tunable operational parameters, called **PARAMS**

The script file which defines the operations that are to be carried out, called **run_script.sh**

The TEMPLATE file

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

Reparametrize.Pseudos T       # Options for more accuracy
Restricted.Radial.Grid F
PAO.SplitTailNorm T

MeshCutoff  200 Ry

%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

DM.Number.Pulay 4

#
# Full template for Basis parameters
#
Basis.Pressure 0.3 GPa         # As in Anglada et al
PAO.SoftDefault    T           # Global soft-confinement options
PAO.SoftPotential 50.0 Ry
PAO.SoftInnerRadius 0.70
#
PAO.BasisType    split
PAO.EnergyShift 0.1 eV
#
# Specify only split-norm parameters
%block PAO.Basis
O     2
 n=2    0    2  S $spl_O
   0.0    0.0
  1.000    1.000
 n=2    1    2  S $spl_O  P 1
   0.0    0.0
   1.000    1.000
H     1
 n=1    0    2  S $spl_H  P 1
   0.0 0.0
   1.000    1.000
%endblock PAO.Basis
```

For the meaning of these input variables, read the Siesta Manual

In this particular example

A soft confinement potential will be used for default for all the atom shells

Default value for the prefactor of the soft confinement (for all atoms and shells)

Default value for the inner radius of the soft confinement (for all atoms and shells): a fraction of the outer confinement radius determined by the energy shift

Some variables are not defined, but kept as variables (with a leading $). Those are the variables that will be optimized in the Simplex minimization. They might change, depending on your particular problem

The TEMPLATE file

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

Reparametrize.Pseudos T       # Options for more accuracy
Restricted.Radial.Grid F
PAO.SplitTailNorm T

MeshCutoff  200 Ry

%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

DM.Number.Pulay 4

#
# Full template for Basis parameters
#
Basis.Pressure 0.3 GPa         # As in Anglada et al
PAO.SoftDefault    T           # Global soft-confinement options
PAO.SoftPotential 50.0 Ry
PAO.SoftInnerRadius 0.70
#
PAO.BasisType    split
PAO.EnergyShift 0.1 eV
#
# Specify only split-norm parameters
%block PAO.Basis
O     2
 n=2    0    2  S $spl_O
   0.0    0.0
  1.000    1.000
 n=2    1    2  S $spl_O  P 1
   0.0    0.0
   1.000    1.000
H     1
 n=1    0    2  S $spl_H  P 1
   0.0 0.0
   1.000   1.000
%endblock PAO.Basis
```

The VARS file

```
spl_O    0.05 0.80   0.15
spl_H    0.05 0.80   0.15
```

Both files define the variables that will be optimized in the Simplex minimization

Match the names that you use in the TEMPLATE (without the leading '$') with those in the VARS file

DO NOT leave any blank lines in the VARS (not even at the end).

The TEMPLATE file

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

Reparametrize.Pseudos T       # Options for more accuracy
Restricted.Radial.Grid F
PAO.SplitTailNorm T

MeshCutoff  200 Ry

%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

DM.Number.Pulay 4

#
# Full template for Basis parameters
#
Basis.Pressure 0.3 GPa         # As in Anglada et al
PAO.SoftDefault    T           # Global soft-confinement options
PAO.SoftPotential 50.0 Ry
PAO.SoftInnerRadius 0.70
#
PAO.BasisType    split
PAO.EnergyShift 0.1 eV
#
# Specify only split-norm parameters
%block PAO.Basis
O     2
 n=2    0    2  S $spl_O
   0.0   0.0
  1.000   1.000
 n=2    1    2  S $spl_O  P 1
   0.0   0.0
   1.000   1.000
H     1
 n=1    0    2  S $spl_H  P 1
   0.0 0.0
   1.000   1.000
%endblock PAO.Basis
```

The VARS file

```
spl_O    0.05 0.80  0.15
spl_H    0.05 0.80  0.15
```

Minimum value for this variable — Maximum value for this variable — Starting value (optional)

If the starting value is missing, a random starting point in the appropriate range is chosen.

The PARAMS file

Tunable operational parameters for the optimization algorithm are defined here

it uses a two-level approach, with amoeba-like iterations followed by periodic restarts with new simplex hyper-tetrahedra of progressively smaller sizes

Initial criteria

```
0.4     # Initial value of lambda (size of simplex)
0.5     # Factor by which lambda decreases at each macro step
0.01    # Minimimum value of lambda
40      # Maximum number of simplex iterations per macro step
1.0e-5  # Fractional tolerance in function value
```

Defines the size of the initial hyper-tetrahedron

The PARAMS file

Tunable operational parameters for the optimization algorithm are defined here

it uses a two-level approach, with amoeba-like iterations followed by periodic restarts with new simplex hyper-tetrahedra of progressively smaller sizes

Stopping criteria

```
0.4     # Initial value of lambda (size of simplex)
0.5     # Factor by which lambda decreases at each macro step
0.01    # Minimimum value of lambda
40      # Maximum number of simplex iterations per macro step
1.0e-5  # Fractional tolerance in function value
```

a minimum fractional size for the hyper-tetrahedron.

and

a fractional energy tolerance (compared to the differences between the highest and lowest energies of the vertices)

To run an optimization

```
$ ~/siesta/Util/Optimizer/simplex > simplex.out &
```

The driver program (simplex) call the script run_script.sh
The script can perform substitutions on suitable template files.
The user has nothing to do at this level.

```
$ tail -f simplex.out
```

```
spl_O  0.050000  0.800000  0.150000
spl_H  0.050000  0.800000  0.150000
 0.4 0.5 0.01
 40 0.00001
 Start of an amoeba cycle: [lambda=  0.4 ]
 Points in simplex and values:
 0.15 0.15  ---  -464.7838495010844
 0.45000000000000007 0.15  ---  -464.3665168358278
 0.15 0.45000000000000007  ---  -465.0905924273053
 Fractional dispersion:  0.0015580613333552446
 New point:  0.05 0.45  ---  -465.07625624190047
 Fractional dispersion:  0.00065975127907546556
 New point:  0.050000000000000044 0.75  ---  -464.92281931818854
 New point:  0.07500000000000004 0.6  ---  -465.1367269144882
 Fractional dispersion:  0.0001300146819764302
 New point:  0.17500000000000013 0.6  ---  -465.11752843408397
 Fractional dispersion:  0.0000991897060506166
 New point:  0.10000000000000026 0.7499999999999998  ---  -464.9204366506313
 New point:  0.13750000000000007 0.525  ---  -465.1370675344798
 Fractional dispersion:  0.00004200807065185912
 New point:  0.05 0.5249999999999997  ---  -465.1220064587157
 New point:  0.07812500000000004 0.5437499999999997  ---  -465.14395320562153
 Fractional dispersion:  0.000015535722256247654
 ...
```

First: it dumps the information of the VARS and PARAMS files

Second: it builds the initial simplex and computes values of the energies at the vertices of the initial simplex

Third: it computes the fractional energy tolerance and checks for the convergence criteria

Fourth: if not converged, an amoeba-like iterations is performed, searching for the minimum and producing simplex hyper-tetrahedra of progressively smaller sizes

At the end of a successful minimization: the final.sed file

It contains the optimized values of the parameters

$ more final.sed

```
s/$spl_O/.1007813/g
s/$spl_H/.5671875/g
```

They can be found also at the end of the output file after running simplex

$ tail simplex.out

```
...
Lambda <  0.01 . Convergence presumably reached
Final (average) point:
Final (average)spl_O : 0.09453125000000007
Final (average)spl_H : 0.5703124999999999
Final (actual minimum):
Finalspl_O : 0.10078125000000007
Finalspl_H : 0.5671874999999998
```