Introduction to run Siesta

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

http://www.icmab.es/siesta

To run Siesta you need

1. Access to the executable file

2. An input file

   Written in Flexible Data Format (A. García and J. M. Soler)

3. An pseudopotential file for each kind of element in the input file

   Unformatted binary (.vps)

   Formatted ASCII (.psf) (more transportable and easy to read)

The main input file contains

1. Physical data of the system

2. Variables to control the approximations

3. Written in Flexible Data Format (fdf),
a language developed by A. García and J. M. Soler

Characteristics of the Flexible Data Format language FDF- I

Data can be given in any order

Data can be omitted in favour of default values

Syntax: 'data label' followed by its value

Character string:	SystemLabel	h2o
Integer:	NumberOfAtoms	3
Real:	PAO.SplitNorm	0.15
Logical:	SpinPolarized	.false.
Physical magnitudes:	LatticeConstant	5.43 Ang

Characteristics of the Flexible Data Format language FDF- II

Labels are case insensitive and characters -_. are ignored
LatticeConstant is equivalent to lattice_constant

Text following # are comments

Logical values: T , .true. , true , yes
F , .false. , false , no

Character strings, NOT in apostrophes

Complex data structures: blocks

```
%block label
…
%endblock label
```

Characteristics of the Flexible Data Format language FDF- III

Physical magnitudes: real number followed by its units

Many physical units are recognized for each magnitude

(Length: m, cm, nm, Ang, bohr)

Automatic conversion to the ones internally required.

Allows to 'include' other FDF files or redirect the search to another file

```
block PAO.Basis                          < BaTiO3basis.fdf
AtomicCoordinatesAndAtomicSpecies        < Interface.coor
```

Clasification of the basic input variables

1. General system descriptors
2. Structural and geometrical variables
3. Basis set generation
4. Variables to control the convergence of the results
5. Method to solve the Hamiltonian
6. Control of the self-consistent cycle
7. Structural relaxation or molecular dynamics
8. Analysis of the results
9. Parallelization

General system descriptors

SystemName: descriptive name of the system

```
SystemName   Si bulk, diamond structure
```

If properly updated, this variable might contain very useful information to know what has been run

SystemLabel: nickname of the system to name output files

```
SystemLabel   Si
```

(After a succesful run, you should have files like

Si.DM :	Density matrix
Si.XV:	Final positions and velocities
Si.bands:	Electronic band structure
Si.DOS:	Total density of states

...and many more, depending on your requests)

Structural and geometrical variables: number of atoms and species in the simulation box

NumberOfAtoms: number of atoms in the simulation box

```
NumberOfAtoms     2
```

NumberOfSpecies: number of different atomic species

```
NumberOfSpecies   1
```

ChemicalSpeciesLabel: specify the different chemical species

```
%block ChemicalSpeciesLabel
1      14     Si
%endblock ChemicalSpeciesLabel
```

From 1 to NumberOfSpecies

Atomic number of a given species plus a label to identify

ALL THESE VARIABLES ARE MANDATORY

Structural and geometrical variables: lattice constant and lattice vectors

LatticeConstant: real length to define the scale of the lattice vectors

```
LatticeConstant        5.43 Ang
```

LatticeParameters: Crystallograhic way

```
%block LatticeParameters
  1.0  1.0  1.0  60.  60.  60.
%endblock LatticeParameters
```

Three vector modules (units of LatticeConstant)

Three angles between vectors (degrees)

LatticeVectors: read as a matrix, each vector being a line (units of LatticeConstant)

```
%block LatticeVectors
  0.0   0.5   0.5
  0.5   0.0   0.5
  0.5   0.5   0.0
%endblock LatticeVectors
```

Structural and geometrical variables: atomic coordinates

AtomicCoordinatesFormat: format of the atomic positions in input:

- Bohr: cartesian coordinates, in bohrs
- Ang: cartesian coordinates, in Angstroms
- ScaledCartesian: cartesian coordinates, units of the lattice constant
- Fractional: referred to the lattice vectors

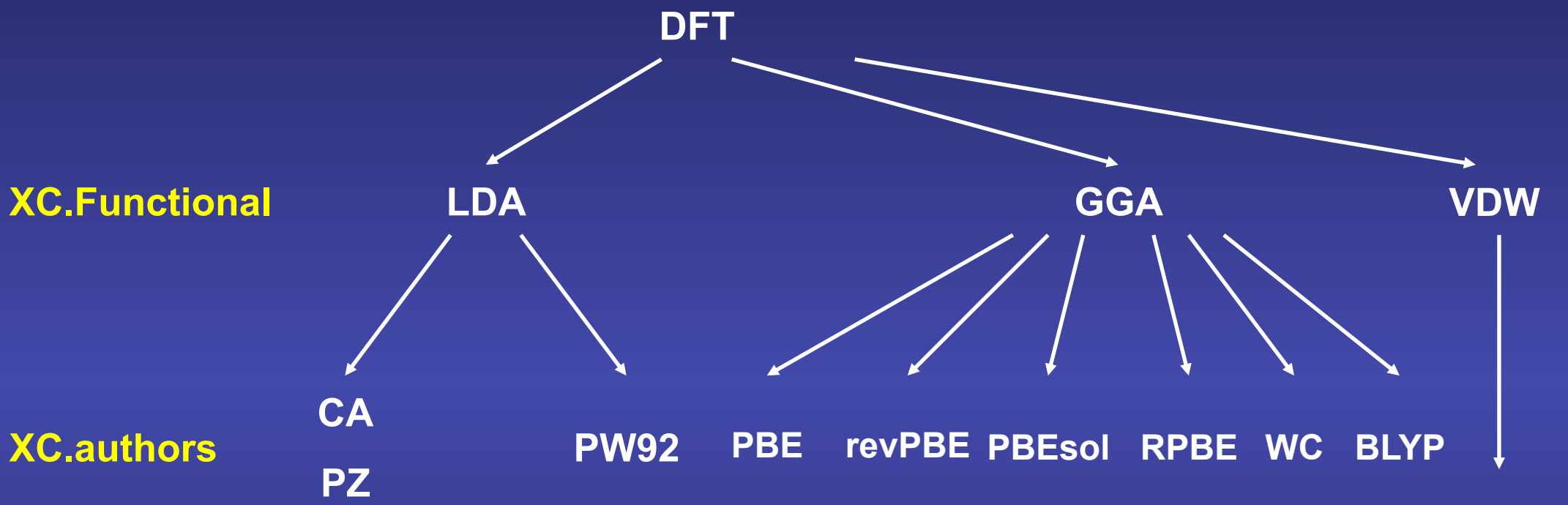
```
AtomicCoordinatesFormat      Fractional
```

AtomicCoordinatesAndAtomicSpecies:

```
%block AtomicCoordinatesAndAtomicSpecies
  0.00   0.00   0.00   1
  0.25   0.25   0.25   1
%endblock AtomicCoordinatesAndAtomicSpecies
```

As many lines as atoms in the simulation box

Variables to control de convergence of the results: the exchange and correlation functional



SpinPolarized

DFT ≡ Density Functional Theory

LDA ≡ Local Density Approximation

GGA ≡ Generalized Gradient Approximation

CA ≡ Ceperley-Alder

PZ ≡ Perdew-Zunger

PW92 ≡ Perdew-Wang-92

PBE ≡ Perdew-Burke-Ernzerhof

WC ≡ Wu-Cohen

BLYP ≡ Becke-Lee-Yang-Parr

Variables to chose the method to solve the Hamiltonian

From the atomic coordinates and the unit cell, the code computes the Hamiltonian (H) and Overlap (S) matrices

$$S_{\nu\mu} = \langle \phi_\nu \mid \phi_\mu \rangle = \int d\vec{r}\ \phi_\nu^* (\vec{r})\, \phi_\mu (\vec{r})$$

$$H_{\nu\mu} = \langle \phi_\nu \mid \hat{H} \mid \phi_\mu \rangle = \int d\vec{r}\ \phi_\nu^* (\vec{r})\, \hat{H} \phi_\mu (\vec{r})$$

This is always done with operations that scale linearly with the size of the system (Order-N)

Then, we have to the secular equation,

$$\sum_\mu (H_{\nu\mu} - E_i S_{\nu\mu})\, c_{\mu i} = 0$$

Once the hamiltonian and the overlap matrices are build, we have to solve the one-particle Kohn-Sham equations

$$\left[H \right]\left[C \right] = E_{n\vec{k}} \left[S \right]\left[C \right]$$

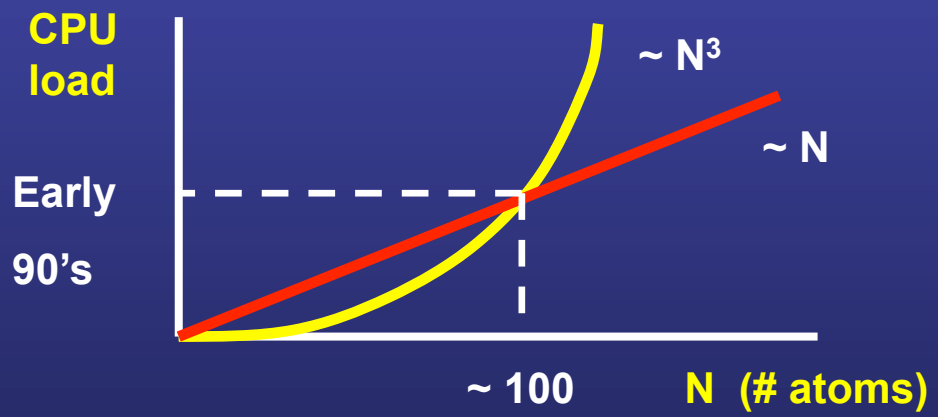
Order-N **Order-N^3**

Minimization of an energy functional

Not efficient for metals or “dirty” gap systems

Standard diagonalization techniques

Both eigenvectors and eigenvalues available



SolutionMethod **Order-N** **diagon**

The Kohn-Sham equations must be solved self-consistently
The potential (input) depends on the density (output)

Initial guess

$$n^{\uparrow}(\vec{r}), n^{\downarrow}(\vec{r})$$

Sum of charge densities of isolated spherical atoms

Calculate effective potential

$$V_{eff}^{\sigma}(\vec{r}) = V_{ext}(\vec{r}) + V_{Hartree}[n] + V_{xc}^{\sigma}[n^{\uparrow}, n^{\downarrow}]$$

Solve the KS equation

$$\left[-\frac{1}{2}\nabla^2 + V_{eff}^{\sigma}(\vec{r})\right]\psi_i^{\sigma}(\vec{r}) = \varepsilon_i^{\sigma}\psi_i^{\sigma}(\vec{r})$$

Compute electron density

$$n^{\sigma}(\vec{r}) = \sum_i f_i^{\sigma}\left|\psi_i^{\sigma}(\vec{r})\right|^2$$

The density matrix, a basic ingredient of SIESTA

Expansion of the eigenvectors in a basis of localized atomic orbitals

$$\psi_i(\vec{r}) = \sum_\mu \phi_\mu(\vec{r})\, c_{\mu i}$$

where the coefficients $c_{\mu i} = \langle \tilde{\phi}_\mu \mid \psi_i \rangle$, and $\tilde{\phi}_\mu$ are the dual orbital of ϕ_μ : $\langle \tilde{\phi}_\mu \mid \phi_\nu \rangle = \delta_{\mu\nu}$

The electron density is given by

$$\rho(\vec{r}) = \sum_i n_i \mid \psi_i(\vec{r}) \mid^2$$

n_i: Occupation of state ψ_i

Inserting the expansion into the definition of the density

$$\rho(\vec{r}) = \sum_{\mu\nu} \rho_{\mu\nu} \phi^*_\nu(\vec{r})\, \phi_\mu(\vec{r})$$

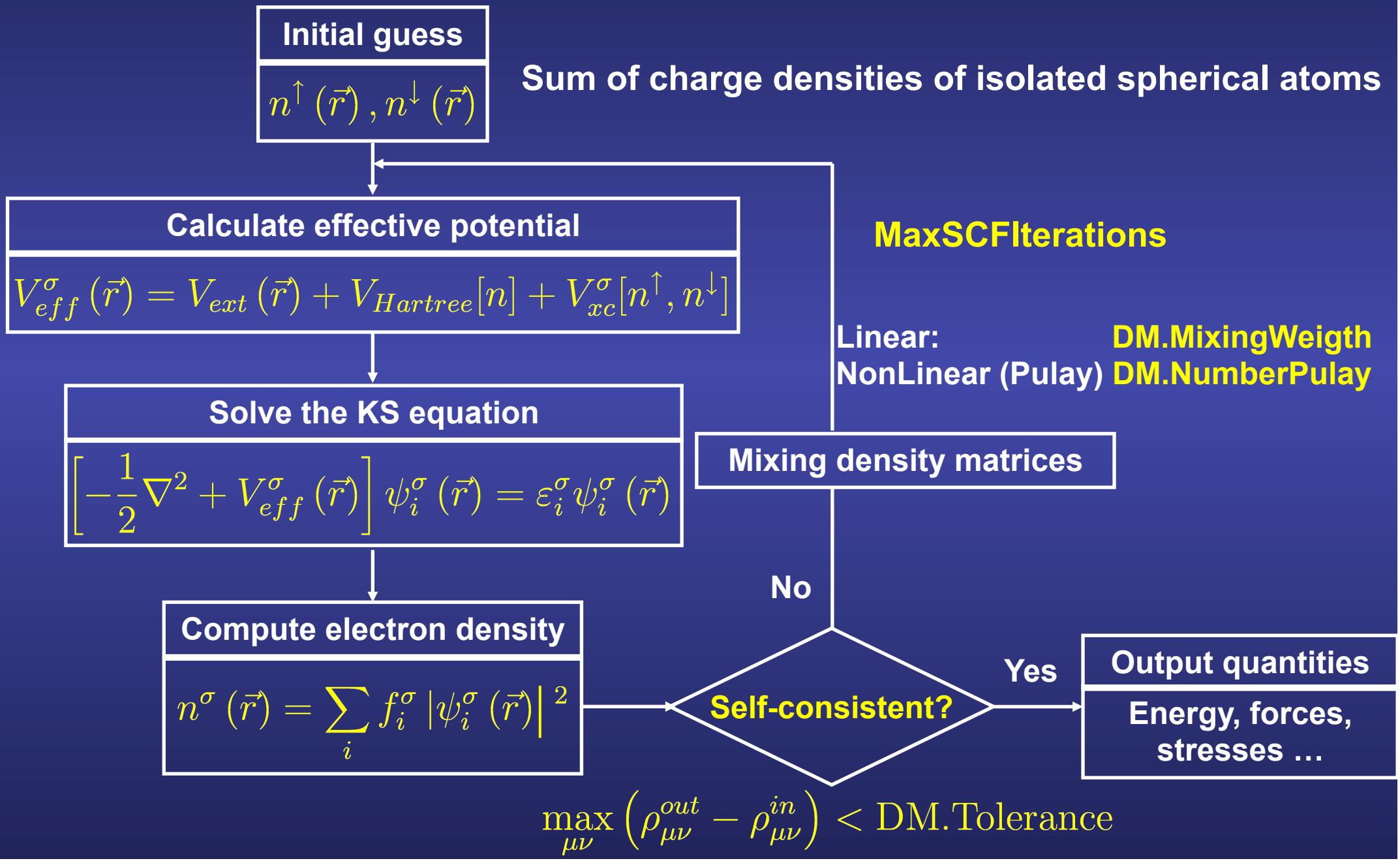
where, with $c_{i\nu} \equiv c^*_{\nu i}$, the **density matrix** is defined

$$\rho_{\mu\nu} = \sum_i c_{\mu i} n_i c_{i\nu}$$

Control convergence SCF

Restart calculations

The Kohn-Sham equations must be solved self-consistently
The potential (input) depends on the density (output)



How to run the serial version of Siesta

To run the serial version:

[path]siesta < myinput.fdf > myoutput

If you want to run the job in background, add an &

[path]siesta < myinput.fdf > myoutput &

To see the information dumped in the

output file during the run:

tail –f myoutput

Output: the header

```
Siesta Version:                                          siesta-3.0-b
Architecture  : g95-nolibs
Compiler flags: g95 -O2 -Wall
SERIAL version

* Running in serial mode
>> Start of run:  12-MAR-2010  16:30:38

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

Information about:

- **The version of Siesta**
- **The compiler**
- **Compilation flags**
- **Mode (serial or parallel)**
- **Date and time when the run starts**

Useful to reproduce the results of a simulation

Output: dumping the input file

```
************************** Dump of input data file ****************************
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
************************** End of input data file *****************************
```

Exact copy of the fdf input file

Useful to reproduce the results of a simulation

Output: processing the input

```
siesta: ******************** Simulation parameters ****************************
siesta:
siesta: The following are some of the parameters of the simulation.
siesta: A complete list of the parameters used, including default values,
siesta: can be found in file fdf.log
siesta:
redata: Non-Collinear-spin run           =     F
redata: SpinPolarized (Up/Down) run      =     F
redata: Number of spin components        =     1
redata: Long output                      =     F
redata: Number of Atomic Species         =        2
redata: Charge density info will appear in .RHO file
redata: Write Mulliken Pop.              =     NO
redata: Mesh Cutoff                      =   300.0000  Ry
redata: Net charge of the system         =     0.0000 |e|
redata: Max. number of SCF Iter          =       50
redata: Mixing is linear
redata: Mix DM in first SCF step ?       =     F
redata: Write Pulay info on disk?        =     F
redata: New DM Mixing Weight             =     0.2500
redata: New DM Occupancy tolerance       = 0.000000000001
redata: No kicks to SCF
redata: DM Mixing Weight for Kicks       =     0.5000
redata: DM Tolerance for SCF             =     0.000100
redata: Require Energy convergence for SCF =     T
redata: DM Energy tolerance for SCF      =     0.000010 eV
redata: Require Harris convergence for SCF =     F
redata: DM Harris energy tolerance for SCF =     0.000100 eV
redata: Using Saved Data (generic)   =     F
redata: Use continuation files for DM    =     F
redata: Neglect nonoverlap interactions  =     F
redata: Method of Calculation            =     Diagonalization
redata: Divide and Conquer               =     T
redata: Electronic Temperature           =     0.0019  Ry
redata: Fix the spin of the system       =     F
redata: Dynamics option                  =     Verlet MD run
redata: Initial MD time step             =        1
redata:   Final MD time step             =        1
redata: Length of MD time step           =     1.0000  fs
redata: Initial Temperature of MD run    =     0.0000  K
redata: Perform a MD quench              =     F
redata: ***********************************************************************
```

The input file is digested

Siesta prints out the value for some variables (some of them might take the default variable)

A complete list of the parameters used, including default values, can be found in the file fdf.log

Output: cell, coordinates and k-sampling Molecules

```
siesta: Atomic coordinates (Bohr) and species
siesta:      0.00000   0.00000   0.00000  1        1
siesta:      1.43052   1.10738   0.00000  2        2
siesta:     -1.43052   1.10738   0.00000  2        3

siesta: Automatic unit cell vectors (Ang):
siesta:    7.286412    0.000000    0.000000
siesta:    0.000000    5.746952    0.000000
siesta:    0.000000    0.000000    5.621012

siesta: System type = molecule

...

 Kpoints in:  1 . Kpoints trimmed:  1

siesta: k-grid: Number of k-points =     1
siesta: k-grid: Cutoff (effective) =     2.811 Ang
siesta: k-grid: Supercell and displacements
siesta: k-grid:    1   0   0      0.000
siesta: k-grid:    0   1   0      0.000
siesta: k-grid:    0   0   1      0.000
```

Output: cell, coordinates and k-sampling
Bulk solids

```
coor:   Atomic-coordinates input format  =     Fractional

siesta: Atomic coordinates (Bohr) and species
siesta:      0.00000   0.00000   0.00000  1        1
siesta:      2.56530   2.56530   2.56530  1        2

siesta: System type = bulk

...

outcell: Unit cell vectors (Ang):
        0.000000    2.715000    2.715000
        2.715000    0.000000    2.715000
        2.715000    2.715000    0.000000

outcell: Cell vector modules (Ang)   :    3.839590    3.839590    3.839590
outcell: Cell angles (23,13,12) (deg):     60.0000     60.0000     60.0000
outcell: Cell volume (Ang**3)        :     40.0258

...

 Kpoints in:  108 . Kpoints trimmed:  108

siesta: k-grid: Number of k-points =   108
siesta: k-grid: Cutoff (effective) =    11.519 Ang
siesta: k-grid: Supercell and displacements
siesta: k-grid:    6   0   0      0.500
siesta: k-grid:    0   6   0      0.500
siesta: k-grid:    0   0   6      0.500
```

Output: First Molecular Dynamic (or Conjugate Gradient) step

```
siesta:                  ==============================
                             Begin MD step =      1
                         ==============================

outcell: Unit cell vectors (Ang):
        7.286412    0.000000    0.000000
        0.000000    5.746952    0.000000
        0.000000    0.000000    5.621012

outcell: Cell vector modules (Ang)   :    7.286412    5.746952    5.621012
outcell: Cell angles (23,13,12) (deg):     90.0000     90.0000     90.0000
outcell: Cell volume (Ang**3)        :    235.3780
New_DM. Step:     1
Initializing Density Matrix...

InitMesh: MESH =    80 x    60 x    60 =      288000
InitMesh: Mesh cutoff (required, used) =   300.000   301.251 Ry

* Maximum dynamic memory allocated =    31 MB

stepf: Fermi-Dirac step function

siesta: Program's energy decomposition (eV):
siesta: Eions   =       815.854478
siesta: Ena     =       175.154321
siesta: Ekin    =       341.667405
siesta: Enl     =       -52.736757
siesta: DEna    =        -0.000002
siesta: DUscf   =         0.000000
siesta: DUext   =         0.000000
siesta: Exc     =      -109.880534
siesta: eta*DQ  =         0.000000
siesta: Emadel  =         0.000000
siesta: Emeta   =         0.000000
siesta: Emolmec =         0.000000
siesta: Ekinion =         0.000000
siesta: Eharris =      -466.520458
siesta: Etot    =      -461.650045
siesta: FreeEng =      -461.650045
```

Output: Self-consistency cycles

```
siesta: iscf   Eharris(eV)      E_KS(eV)   FreeEng(eV)   dDmax  Ef(eV)
siesta:    1     -466.5205     -461.6500     -461.6500  1.4318 -4.4335
timer: Routine,Calls,Time,% = IterSCF        1       4.820  78.39
elaps: Routine,Calls,Wall,% = IterSCF        1       4.821  78.39
siesta:    2     -467.2399     -464.9984     -464.9984  0.2134  0.2175
siesta:    3     -465.9188     -465.2990     -465.2990  0.0603 -1.4873
siesta:    4     -465.8164     -465.4342     -465.4342  0.0237 -1.9527
siesta:    5     -465.8072     -465.5280     -465.5280  0.0121 -2.0840
siesta:    6     -465.8063     -465.5976     -465.5976  0.0072 -2.1198
siesta:    7     -465.8062     -465.6498     -465.6498  0.0048 -2.1283
siesta:    8     -465.8062     -465.6889     -465.6889  0.0033 -2.1293
siesta:    9     -465.8061     -465.7182     -465.7182  0.0024 -2.1287
siesta:   10     -465.8061     -465.7402     -465.7402  0.0017 -2.1279
siesta:   11     -465.8061     -465.7567     -465.7567  0.0012 -2.1273
siesta:   12     -465.8061     -465.7690     -465.7690  0.0009 -2.1269
siesta:   13     -465.8061     -465.7783     -465.7783  0.0007 -2.1266
siesta:   14     -465.8061     -465.7853     -465.7853  0.0005 -2.1264
siesta:   15     -465.8061     -465.7905     -465.7905  0.0004 -2.1262
siesta:   16     -465.8061     -465.7944     -465.7944  0.0003 -2.1262
siesta:   17     -465.8061     -465.7973     -465.7973  0.0002 -2.1261
siesta:   18     -465.8061     -465.7995     -465.7995  0.0002 -2.1261
siesta:   19     -465.8061     -465.8012     -465.8012  0.0001 -2.1260
siesta:   20     -465.8061     -465.8024     -465.8024  0.0001 -2.1260
siesta:   21     -465.8061     -465.8034     -465.8034  0.0001 -2.1260
siesta:   22     -465.8061     -465.8041     -465.8041  0.0001 -2.1260
```

Output: Forces and stress tensor

```
siesta: Atomic forces (eV/Ang):
siesta:      1    0.000000   -0.734675    0.000000
siesta:      2    0.691039    0.367258   -0.000000
siesta:      3   -0.691039    0.367258   -0.000000
siesta: ----------------------------------------
siesta:    Tot    0.000000   -0.000159    0.000000

siesta: Stress tensor (static) (eV/Ang**3):
siesta:    -0.004475    0.000000    0.000000
siesta:    -0.000000   -0.001644   -0.000000
siesta:     0.000000    0.000000   -0.000069
```

WriteForces (logical) write the forces to the output file at the end of every Molecular Dynamic step or relaxation step.

The forces of the last step can be found in the file SystemLabel.FA

Output: Descomposition of the energy

```
siesta: Program's energy decomposition (eV):
siesta: Eions   =       815.854478
siesta: Ena     =       175.154321
siesta: Ekin    =       353.637801
siesta: Enl     =       -63.031215
siesta: DEna    =        -3.294344
siesta: DUscf   =         0.753779
siesta: DUext   =         0.000000
siesta: Exc     =      -113.171984
siesta: eta*DQ  =         0.000000
siesta: Emadel  =         0.000000
siesta: Emeta   =         0.000000
siesta: Emolmec =         0.000000
siesta: Ekinion =         0.000000
siesta: Eharris =      -465.806142
siesta: Etot    =      -465.806121
siesta: FreeEng =      -465.806121

siesta: Final energy (eV):
siesta:      Kinetic =     353.637801
siesta:      Hartree =     385.082216
siesta:   Ext. field =       0.000000
siesta:  Exch.-corr. =    -113.171984
siesta:  Ion-electron =   -1079.842112
siesta:      Ion-ion =     -11.512042
siesta:      Ekinion =       0.000000
siesta:        Total =    -465.806121
```

Output: timer. How many times (and how much time) the code goes through the most significant subroutines

```
elaps: ELAPSED times:
elaps:  Routine       Calls   Time/call    Tot.time        %
elaps:  siesta            1      35.589      35.589   100.00
elaps:  Setup             1       0.539       0.539     1.52
elaps:  bands             1       0.001       0.001     0.00
elaps:  writewave         1       0.003       0.003     0.01
elaps:  KSV_init          1       0.000       0.000     0.00
elaps:  IterMD            1      35.032      35.032    98.44
elaps:  hsparse           1       0.001       0.001     0.00
elaps:  overlap           1       0.783       0.783     2.20
elaps:  IterSCF          37       0.856      31.667    88.98
elaps:  kinefsm           2       0.381       0.763     2.14
elaps:  nlefsm            2       1.051       2.102     5.91
elaps:  DHSCF            38       0.822      31.244    87.79
elaps:  DHSCF1            1       0.218       0.218     0.61
elaps:  DHSCF2            1       1.023       1.023     2.87
elaps:  REORD           306       0.003       0.921     2.59
elaps:  POISON           39       0.132       5.160    14.50
elaps:  DHSCF3           38       0.732      27.811    78.14
elaps:  rhoofd           38       0.114       4.332    12.17
elaps:  cellXC           38       0.350      13.302    37.38
elaps:  vmat             38       0.065       2.462     6.92
elaps:  MolMec            2       0.000       0.000     0.00
elaps:  diagon           37       0.001       0.027     0.08
elaps:  r-eigvec         37       0.001       0.022     0.06
elaps:  r-buildHS        37       0.000       0.001     0.00
elaps:  rdiag            37       0.001       0.021     0.06
elaps:  rdiag1           37       0.000       0.001     0.00
elaps:  rdiag2           37       0.000       0.001     0.00
elaps:  rdiag3           37       0.000       0.012     0.04
elaps:  rdiag4           37       0.000       0.001     0.00
elaps:  r-buildD         37       0.000       0.001     0.00
elaps:  DHSCF4            1       1.836       1.836     5.16
elaps:  dfscf             1       1.239       1.239     3.48
elaps:  overfsm           1       0.001       0.001     0.00
elaps:  optical           1       0.001       0.001     0.00

>> End of run:  12-MAR-2010  16:31:13
```

Useful to tune and optimize the performance of Siesta

Saving and reading information: Restarting files

Some information is stored by Siesta to restart simulations from:

	Name of the file	FDF tag to reuse
Density matrix	SystemLabel.DM	DM.UseSaveDM
Atomic positions and velocities	SystemLabel.XV	MD.UseSaveXV
Conjugent gradient history	SystemLabel.CG	MD.UseSaveCG
Localized wave functions (Order-N)	SystemLabel.LWF	ON.UseSaveLWF
		logical variables

EXTREMELY USEFUL TO SAVE LOT OF TIME!

Saving and reading information

Information needed as input for various post-processing programs,

for example, to visualize:

	FDF tag to save files	Name of output file
Total charge density:	SaveRho	SystemLabel.RHO
Deformation charge density:	SaveDeltaRho	SystemLabel.DRHO
Electrostatic potential:	SaveElectrostaticPotential	SystemLabel.VH
Total potential:	SaveTotalPotential	SystemLabel.VT
Local density of states:	LocalDensityOfStates	SystemLabel.LDOS
Charge density contours:	WriteDenchar	SystemLabel.DIM
		SystemLabel.PLD
Atomic coordinates:	WriteCoorXmol	SystemLabel.xyz
	WriteCoorCerius	SystemLabel.xtl
Animation of a molecular dyn:	WriteMDXMol	SystemLabel.ANI
	(logical variables)	