From DFT to MLFT

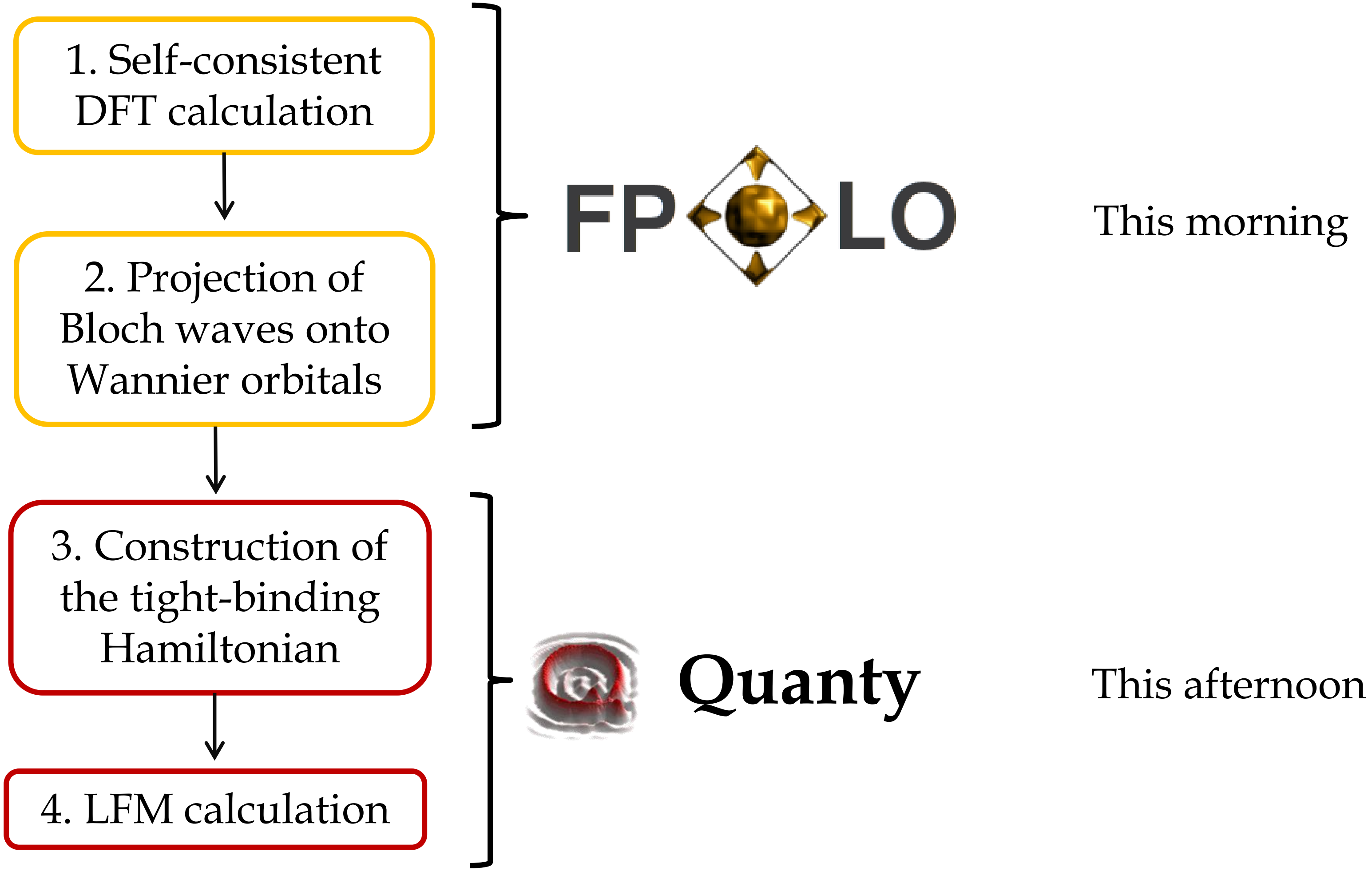
Use of FPLO and Quanty to perform ab-initio ligand field multiplets calculations

N. Bouldi, S. Heinze and M.W. Haverkort

* *Institute for Theoretical Physics, Heidelberg University*

Workshops in Heidelberg, Germany
Hands on spectroscopy calculations of quantum materials
September 27th 2018

“*Ab-initio*” Ligand Field Multiplet: procedure



1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

3. Construction of the tight-binding Hamiltonian

4. LFM calculation

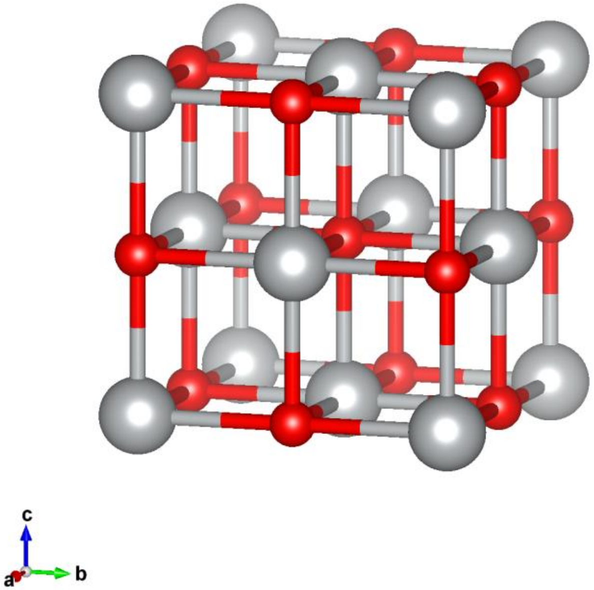
FPLO This morning

Quanty This afternoon

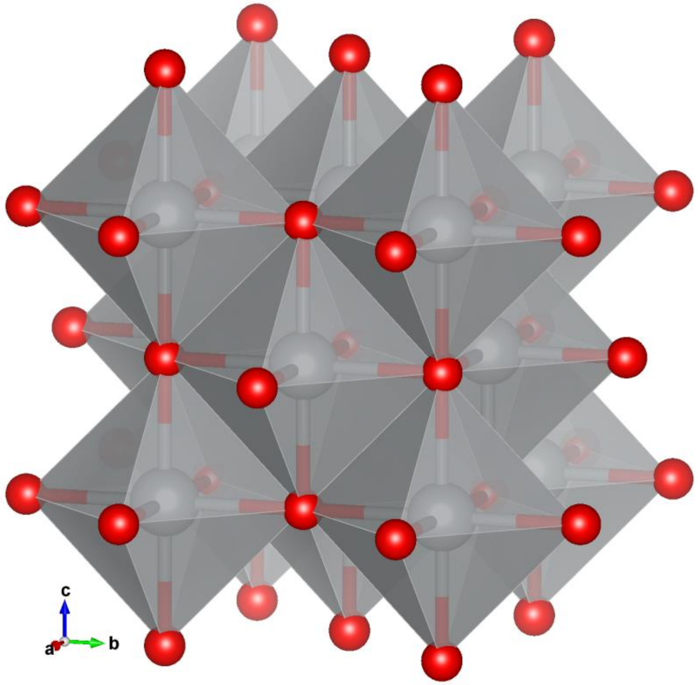
Example I: NiO

NiO: a strongly correlated oxyde

Rock salt structure
Space group $Fm\bar{3}m$
a=4.178 Å

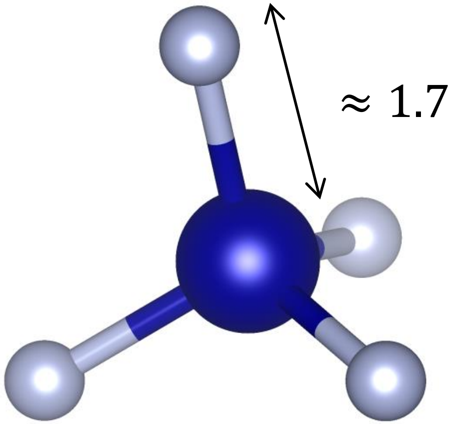


NiO6 octahedra



Example II: molecule CrF_4

CrF_4: Tetrahedral molecule



≈ 1.7 Å (Bond length will be relaxed)

Part 1: Density functional theory calculation : bands, DOS, Wannier functions

"*Ab-initio*" Ligand Field Multiplet: procedure

1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

FPLO

3. Construction of the tight-binding Hamiltonian

4. LFM calculation

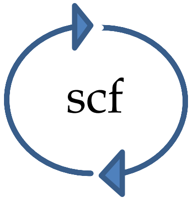
Quanty

FPLO

FPLO

Full-potential local-orbital

- Kohn-Sham DFT - Self consistent



Basis: local orbitals at site **s** in the cell at **T**
(non orthogonal)

- $$u_{n,\mathbf{k}}(r) = \frac{1}{\sqrt{N}} \sum_{\mathbf{T},\mathbf{s},L} \phi_{\mathbf{s},L}(\mathbf{r}-\mathbf{T}-\mathbf{s}) C_{n,\mathbf{k}}^{\mathbf{s},L} e^{i\mathbf{k}.(\mathbf{T}+\mathbf{s})}$$

- Solution of the **exact problem** for valence orbitals (frozen core electrons)

K. Koepernik and H. Eschrig, *Phys. Rev. B* **59**, 1743 (1999).
H. Eschrig, K. Koepernik, and I. Chaplygin, *J. Solid State Chemistry* **176**, 482 (2003).

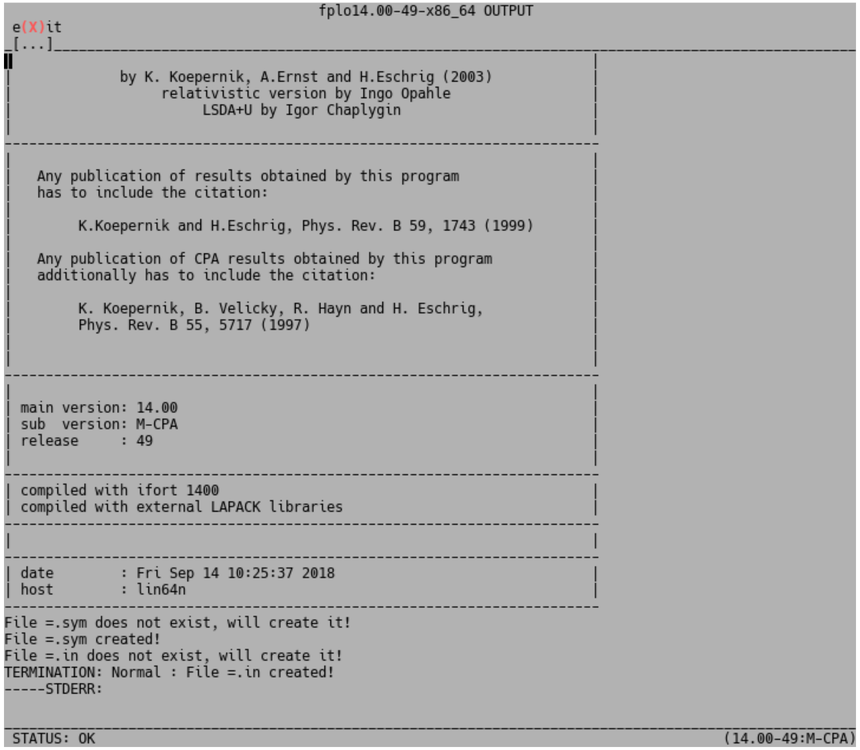
fedit

FPLO includes an input editor: `fedit`

```
$ cd TutorialDFTtoMLFT/NiO && mkdir DFT && cd DFT
```

```
$ fedit14.00-49-x86_64
```

Numbers vary with the version you have installed

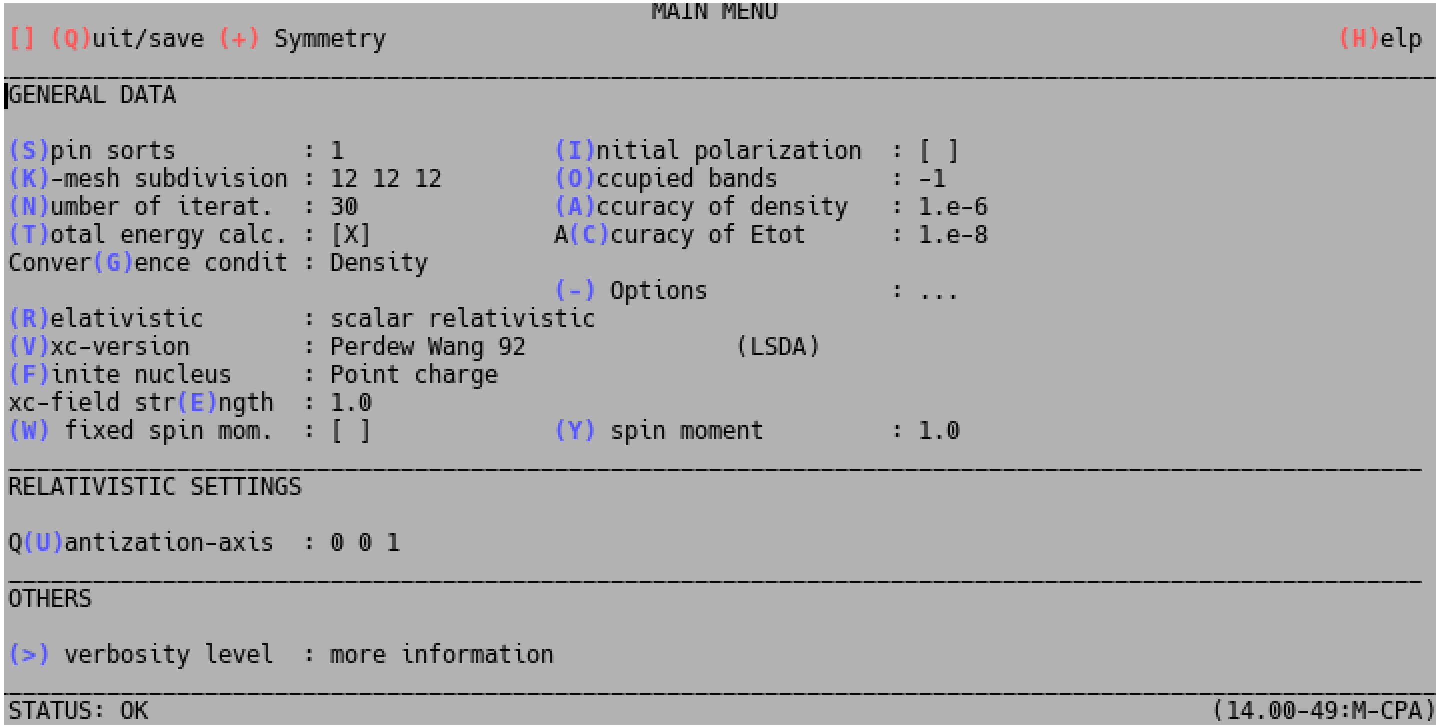


```
                                  fplo14.00-49-x86_64 OUTPUT
 e(X)it
_[...]___________________________________________________________________________________________
|                                                                      |
|             by K. Koepernik, A.Ernst and H.Eschrig (2003)            |
|                  relativistic version by Ingo Opahle                 |
|                       LSDA+U by Igor Chaplygin                       |
|                                                                      |
------------------------------------------------------------------------
|                                                                      |
|   Any publication of results obtained by this program                |
|   has to include the citation:                                       |
|                                                                      |
|        K.Koepernik and H.Eschrig, Phys. Rev. B 59, 1743 (1999)       |
|                                                                      |
|   Any publication of CPA results obtained by this program            |
|   additionally has to include the citation:                          |
|                                                                      |
|        K. Koepernik, B. Velicky, R. Hayn and H. Eschrig,             |
|        Phys. Rev. B 55, 5717 (1997)                                  |
|                                                                      |
|                                                                      |
------------------------------------------------------------------------
|                                                                      |
| main version: 14.00                                                  |
| sub  version: M-CPA                                                  |
| release     : 49                                                     |
|                                                                      |
------------------------------------------------------------------------
| compiled with ifort 1400                                             |
| compiled with external LAPACK libraries                              |
------------------------------------------------------------------------
|                                                                      |
------------------------------------------------------------------------
| date        : Fri Sep 14 10:25:37 2018                               |
| host        : lin64n                                                 |
------------------------------------------------------------------------
File =.sym does not exist, will create it!
File =.sym created!
File =.in does not exist, will create it!
TERMINATION: Normal : File =.in created!
-----STDERR:


_________________________________________________________________________________________________
 STATUS: OK                                                                        (14.00-49:M-CPA)
```

Type **X**

fedit

Main menu



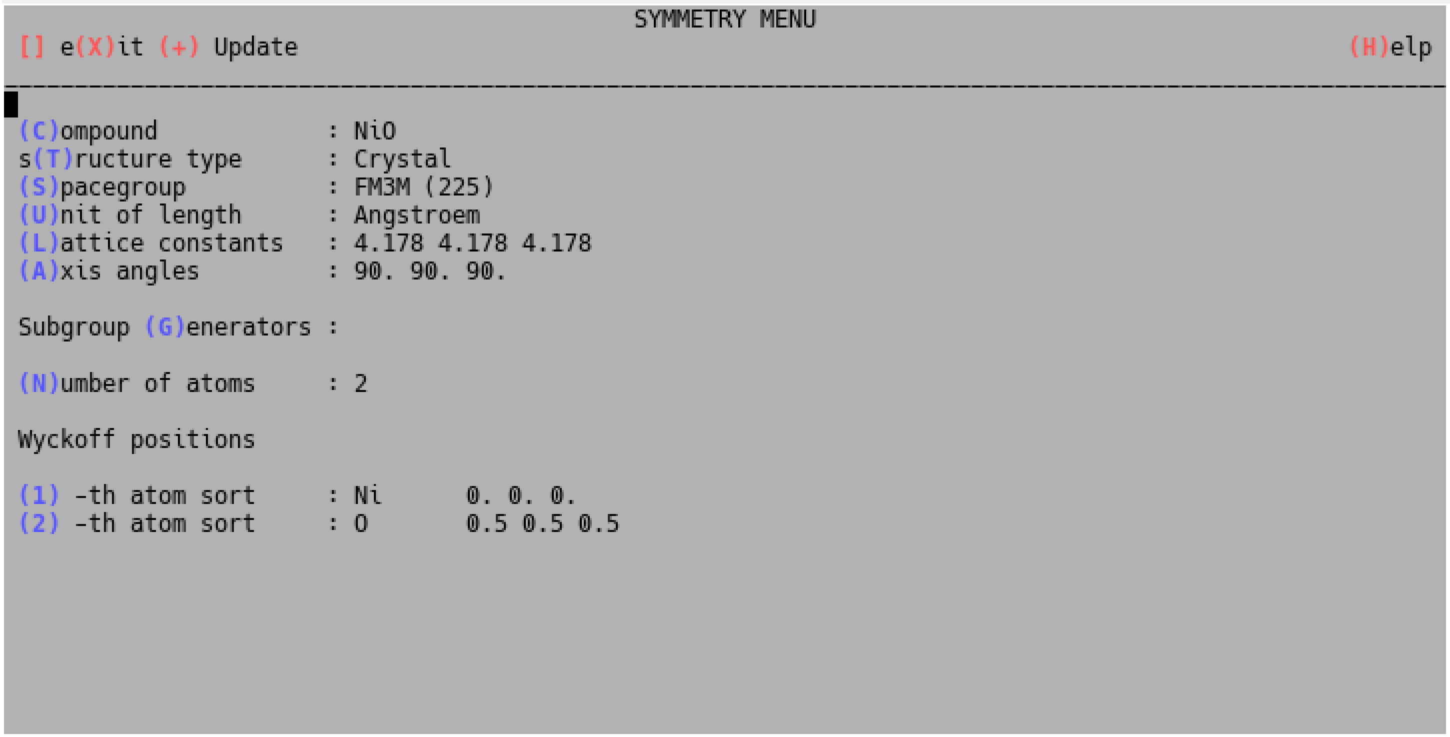
```
                                   MAIN MENU
[] (Q)uit/save (+) Symmetry                                                                    (H)elp
------------------------------------------------------------------------------------------------------
GENERAL DATA

(S)pin sorts          : 1             (I)nitial polarization  : [ ]
(K)-mesh subdivision : 12 12 12       (O)ccupied bands        : -1
(N)umber of iterat.  : 30             (A)ccuracy of density   : 1.e-6
(T)otal energy calc. : [X]            A(C)curacy of Etot      : 1.e-8
Conver(G)ence condit : Density
                                      (-) Options             : ...
(R)elativistic        : scalar relativistic
(V)xc-version         : Perdew Wang 92            (LSDA)
(F)inite nucleus      : Point charge
xc-field str(E)ngth  : 1.0
(W) fixed spin mom.  : [ ]            (Y) spin moment         : 1.0
------------------------------------------------------------------------------------------------------
RELATIVISTIC SETTINGS

Q(U)antization-axis  : 0 0 1
------------------------------------------------------------------------------------------------------
OTHERS

(>) verbosity level  : more information
------------------------------------------------------------------------------------------------------
STATUS: OK                                                                             (14.00-49:M-CPA)
```

Type + to enter symmetry menu

Red keys: select another menu or exit
H opens a help screen
Blue keys: select an entry

fedit

Symmetry



```
                                         SYMMETRY MENU
[] e(X)it (+) Update                                                                         (H)elp
-----------------------------------------------------------------------------------------------------

(C)ompound               : NiO
s(T)ructure type         : Crystal
(S)pacegroup             : FM3M (225)
(U)nit of length         : Angstroem
(L)attice constants      : 4.178 4.178 4.178
(A)xis angles            : 90. 90. 90.

Subgroup (G)enerators :

(N)umber of atoms        : 2

Wyckoff positions

(1) -th atom sort        : Ni      0. 0. 0.
(2) -th atom sort        : O       0.5 0.5 0.5
```

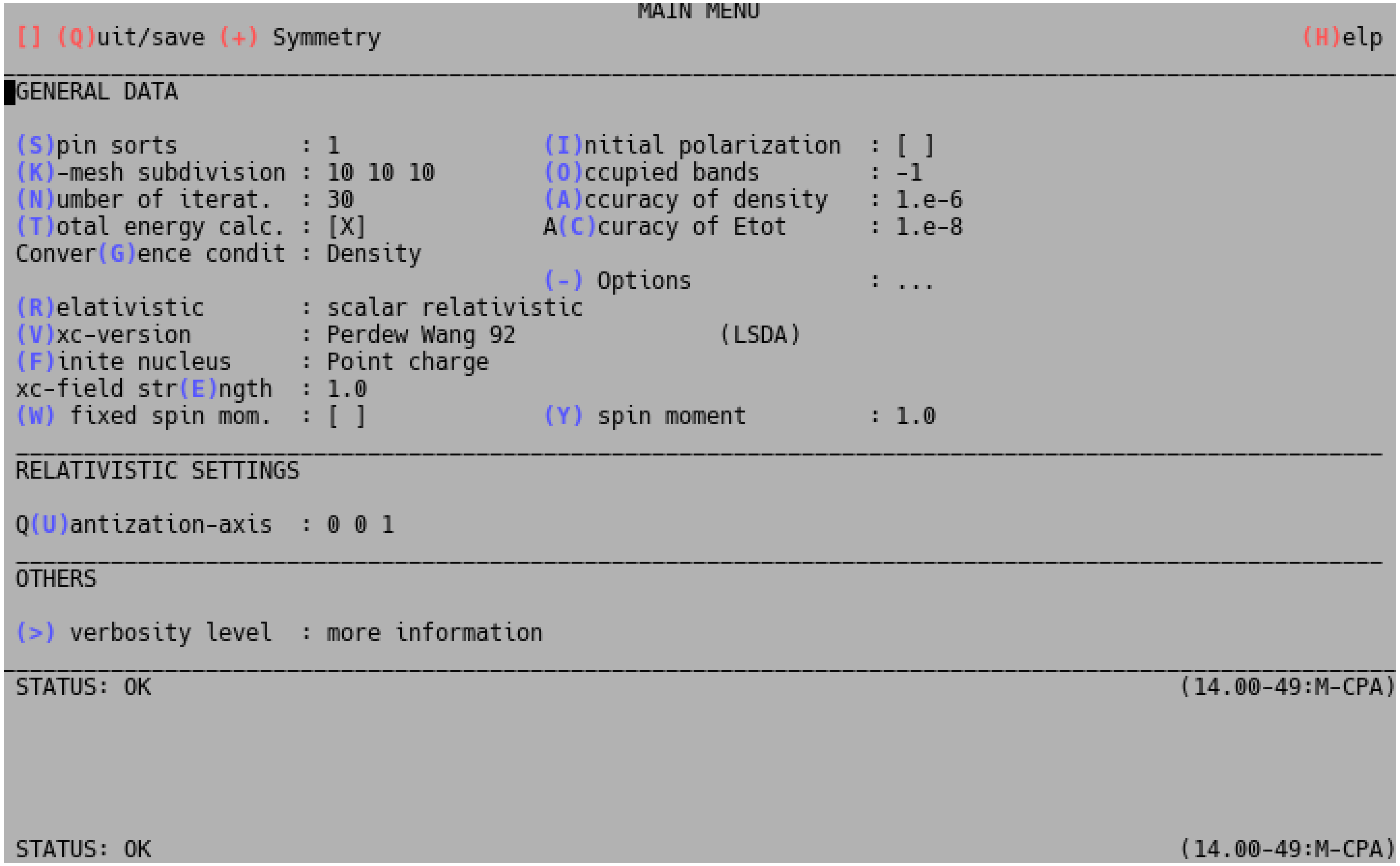
Fill in the lattice structure and chemical composition

Type + to update   Do not forget !!!

then type X X to come back to the main menu

1st scf calculation

Main menu



```
                                  MAIN MENU
[] (Q)uit/save (+) Symmetry                                                                     (H)elp
------------------------------------------------------------------------------------------------------
GENERAL DATA

(S)pin sorts          : 1               (I)nitial polarization  : [ ]
(K)-mesh subdivision : 10 10 10         (O)ccupied bands        : -1
(N)umber of iterat.   : 30              (A)ccuracy of density   : 1.e-6
(T)otal energy calc. : [X]              A(C)curacy of Etot      : 1.e-8
Conver(G)ence condit : Density
                                        (-) Options             : ...
(R)elativistic        : scalar relativistic
(V)xc-version         : Perdew Wang 92            (LSDA)
(F)inite nucleus      : Point charge
xc-field str(E)ngth   : 1.0
(W) fixed spin mom.   : [ ]             (Y) spin moment         : 1.0
------------------------------------------------------------------------------------------------------
RELATIVISTIC SETTINGS

Q(U)antization-axis   : 0 0 1
------------------------------------------------------------------------------------------------------
OTHERS

(>) verbosity level   : more information
------------------------------------------------------------------------------------------------------
STATUS: OK                                                                              (14.00-49:M-CPA)




STATUS: OK                                                                              (14.00-49:M-CPA)
```

Set the number of k-points to 10x10x10

Type Q y to quit fedit

Run FPLO:

```
$ fplo14.00-49-x86_64 | tee out.scflow
```

fplo automatically uses the files created by fedit

1st scf calculation

Output (`out.scflow`) : Iteration progress, Population Analysis, Energy

`=.dens`: electron density; if present in the folder it is used as input by FPLO

Iteration progress:

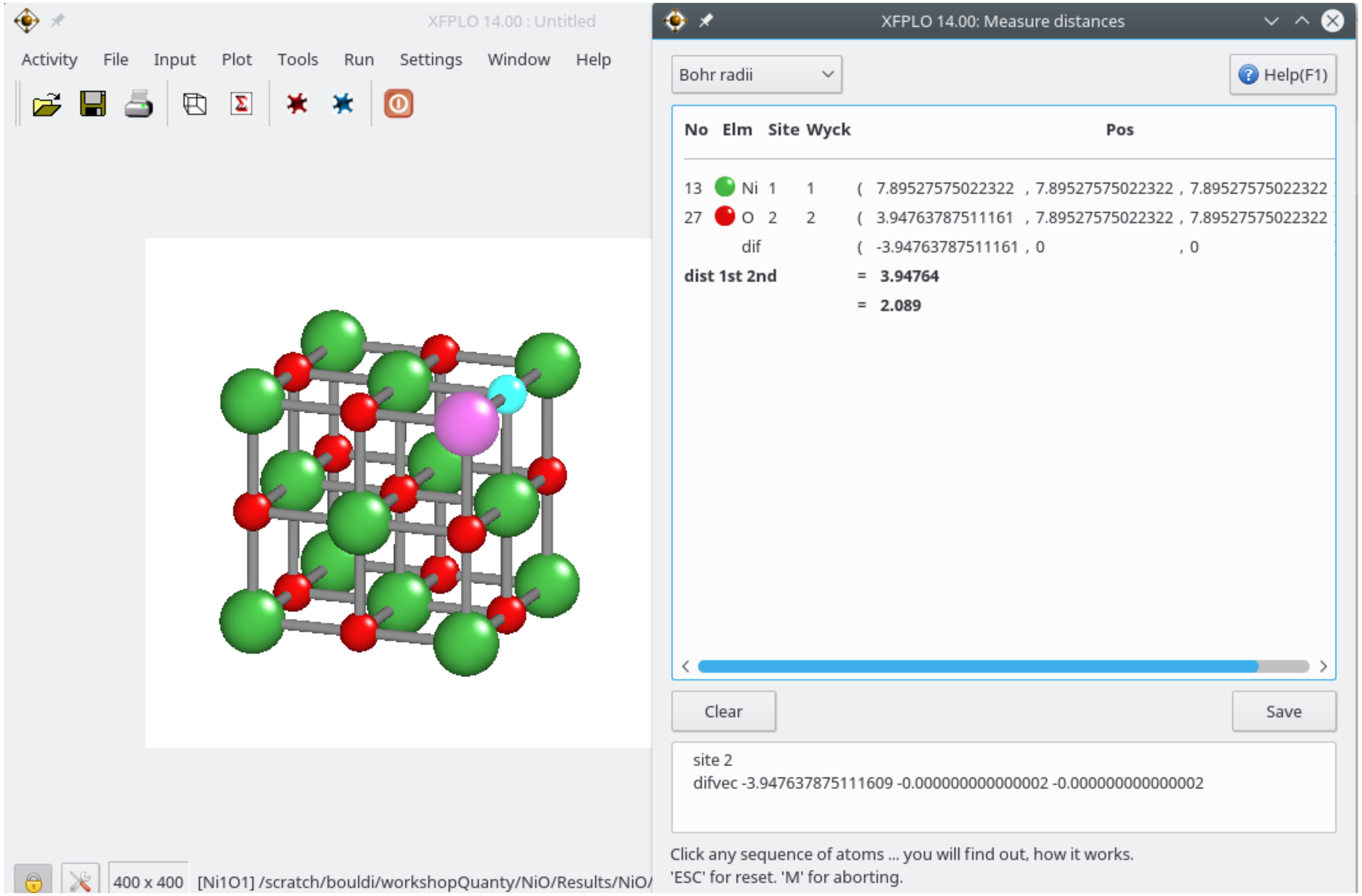
```
$ grep 'last deviation' out.scflow
```

```
SCF: iteration  0  dimension  0  last deviation u=  0.00E+00
SCF: iteration  1  dimension  1  last deviation u=  0.24E+00
SCF: iteration  2  dimension  2  last deviation u=  0.23E+00
SCF: iteration  3  dimension  3  last deviation u=  0.81E-01
SCF: iteration  4  dimension  1  last deviation u=  0.28E-01
SCF: iteration  5  dimension  2  last deviation u=  0.55E-02
SCF: iteration  6  dimension  3  last deviation u=  0.29E-02
SCF: iteration  7  dimension  1  last deviation u=  0.13E-02
SCF: iteration  8  dimension  2  last deviation u=  0.18E-03
SCF: iteration  9  dimension  3  last deviation u=  0.73E-04
SCF: iteration 10  dimension  1  last deviation u=  0.47E-04
SCF: iteration 11  dimension  2  last deviation u=  0.49E-05
SCF: iteration 12  dimension  1  last deviation u=  0.26E-05
SCF: iteration 13  dimension  1  last deviation u=  0.86E-06
CONVERGED
```

Display structure

FPLO includes a graphical interface: `XFPLO`

```
$ xfplo =.in
```



```
No  Elm  Site Wyck                           Pos

13  ● Ni  1    1      (  7.89527575022322  , 7.89527575022322 , 7.89527575022322
27  ● O   2    2      (  3.94763787511161  , 7.89527575022322 , 7.89527575022322
      dif             ( -3.94763787511161 , 0                  , 0
dist 1st 2nd          =  3.94764
                      =  2.089
```

```
site 2
difvec -3.947637875111609 -0.000000000000002 -0.000000000000002
```

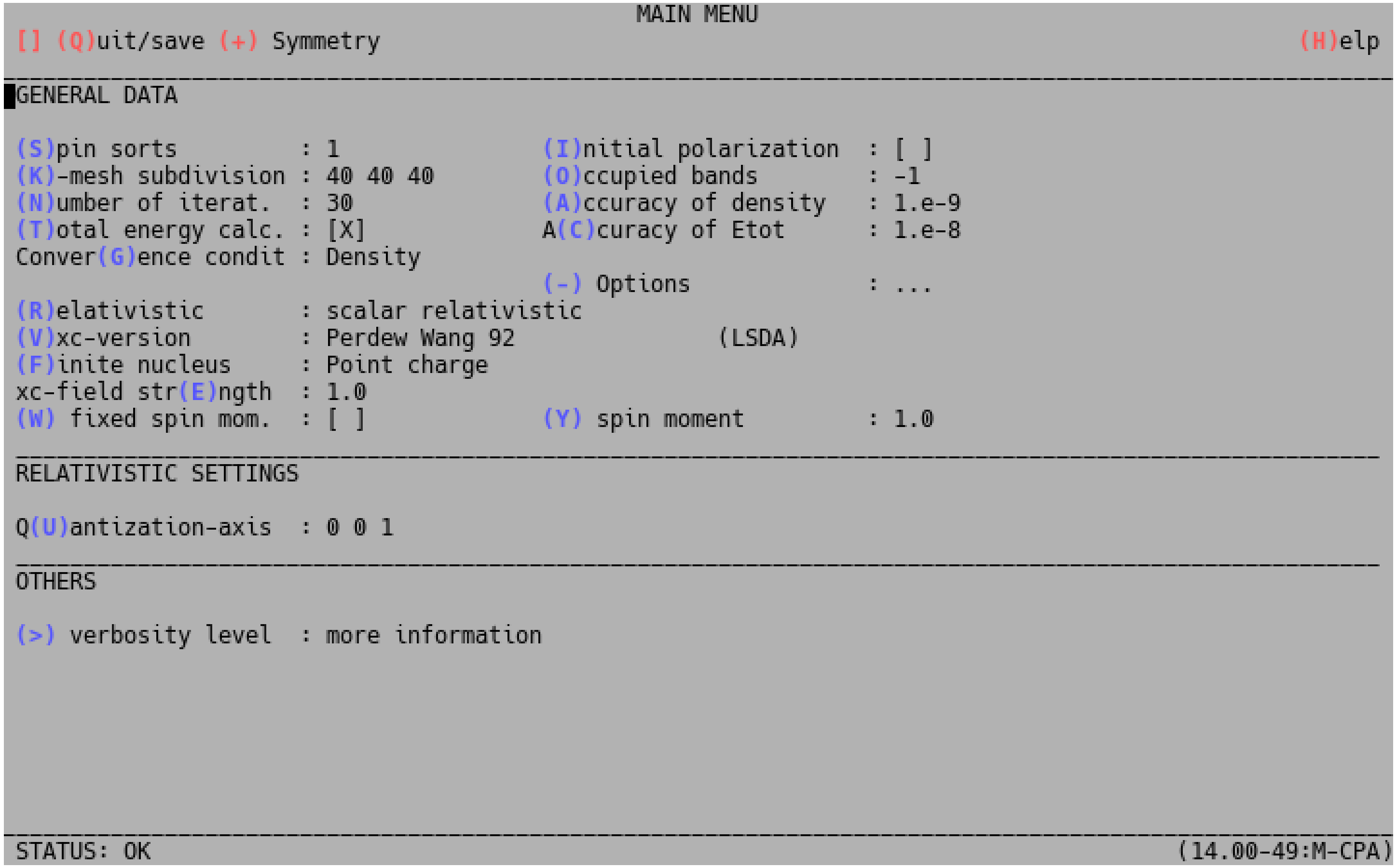
Click any sequence of atoms ... you will find out, how it works.
'ESC' for reset. 'M' for aborting.

400 x 400 [Ni1O1] /scratch/bouldi/workshopQuanty/NiO/Results/NiO/

2nd scf calculation

```
$ fedit14.00-49-x86_64
```

Main menu



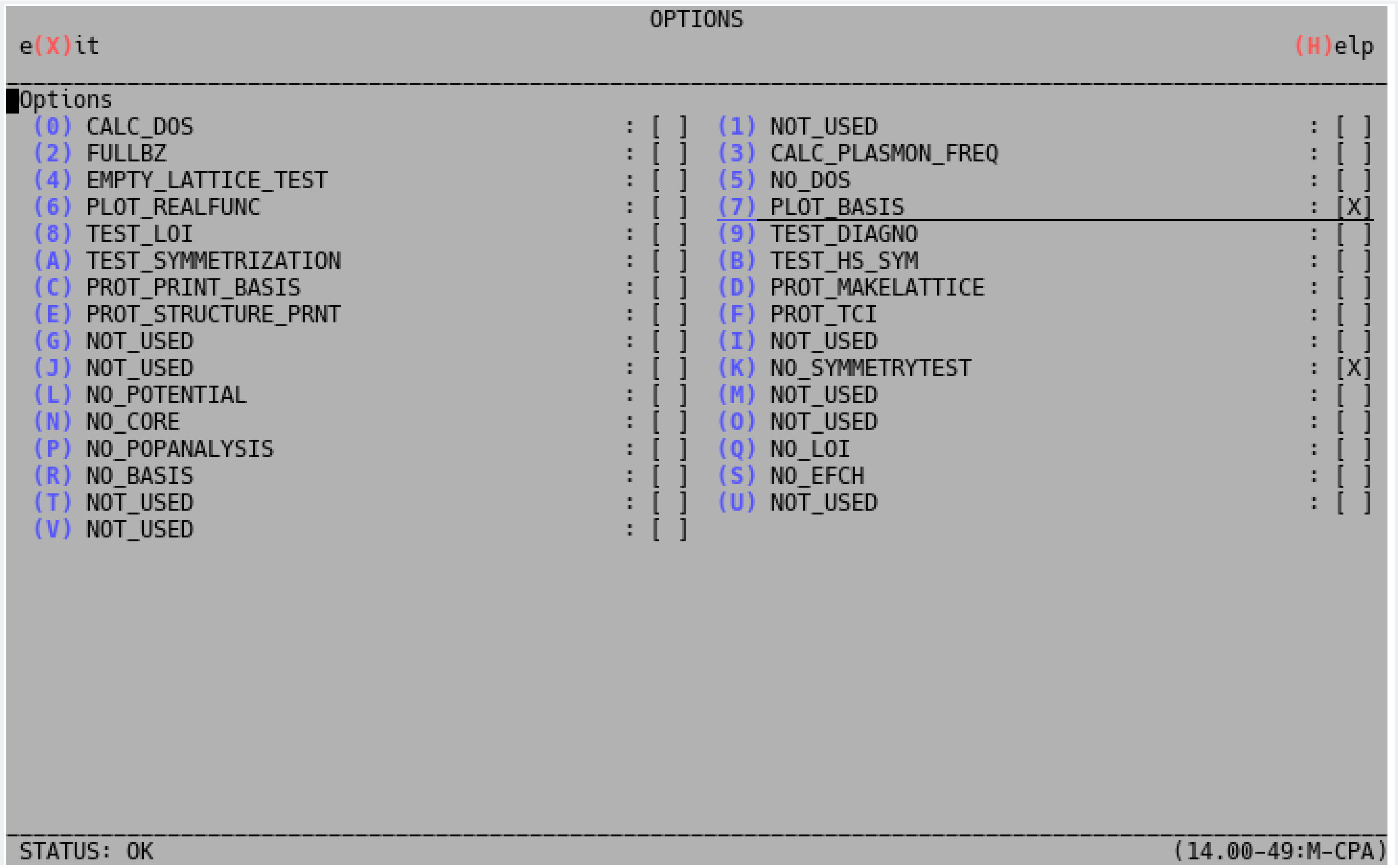
```
                                   MAIN MENU
[] (Q)uit/save (+) Symmetry                                                      (H)elp
----------------------------------------------------------------------------------------
GENERAL DATA

(S)pin sorts          : 1               (I)nitial polarization  : [ ]
(K)-mesh subdivision : 40 40 40         (O)ccupied bands        : -1
(N)umber of iterat.   : 30              (A)ccuracy of density   : 1.e-9
(T)otal energy calc. : [X]              A(C)curacy of Etot      : 1.e-8
Conver(G)ence condit : Density
                                        (-) Options             : ...
(R)elativistic        : scalar relativistic
(V)xc-version         : Perdew Wang 92              (LSDA)
(F)inite nucleus      : Point charge
xc-field str(E)ngth  : 1.0
(W) fixed spin mom.   : [ ]             (Y) spin moment         : 1.0
----------------------------------------------------------------------------------------
RELATIVISTIC SETTINGS

Q(U)antization-axis  : 0 0 1
----------------------------------------------------------------------------------------
OTHERS

(>) verbosity level  : more information




----------------------------------------------------------------------------------------
STATUS: OK                                                                (14.00-49:M-CPA)
```

Increase the number of k-points to 40x40x40
and decrease the accuracy of density to 10^{-9}

Type `-` to enter the Options menu

2nd scf calculation

Options



```
                                       OPTIONS
e(X)it                                                                                  (H)elp
------------------------------------------------------------------------------------------------
Options
  (0) CALC_DOS                              : [ ]  (1) NOT_USED                          : [ ]
  (2) FULLBZ                                : [ ]  (3) CALC_PLASMON_FREQ                 : [ ]
  (4) EMPTY_LATTICE_TEST                    : [ ]  (5) NO_DOS                            : [ ]
  (6) PLOT_REALFUNC                         : [ ]  (7) PLOT_BASIS                        : [X]
  (8) TEST_LOI                              : [ ]  (9) TEST_DIAGNO                       : [ ]
  (A) TEST_SYMMETRIZATION                   : [ ]  (B) TEST_HS_SYM                       : [ ]
  (C) PROT_PRINT_BASIS                      : [ ]  (D) PROT_MAKELATTICE                  : [ ]
  (E) PROT_STRUCTURE_PRNT                   : [ ]  (F) PROT_TCI                          : [ ]
  (G) NOT_USED                              : [ ]  (I) NOT_USED                          : [ ]
  (J) NOT_USED                              : [ ]  (K) NO_SYMMETRYTEST                   : [X]
  (L) NO_POTENTIAL                          : [ ]  (M) NOT_USED                          : [ ]
  (N) NO_CORE                               : [ ]  (O) NOT_USED                          : [ ]
  (P) NO_POPANALYSIS                        : [ ]  (Q) NO_LOI                            : [ ]
  (R) NO_BASIS                              : [ ]  (S) NO_EFCH                           : [ ]
  (T) NOT_USED                              : [ ]  (U) NOT_USED                          : [ ]
  (V) NOT_USED                              : [ ]

------------------------------------------------------------------------------------------------
STATUS: OK                                                                     (14.00-49:M-CPA)
```

Type 7 to save the basis orbitals to disk

Type X Q y to quit

2nd scf calculation

Run FPLO:

```
$ fplo14.00-49-x86_64 | tee out.scf
```

`+fcor.sort.spin:` radial part of the core orbitals
`+fval.sort.spin:` radial part of the valence orbitals
`+fkcor.sort.spin:` kinetic core functions
`+fkval.sort.spin:` kinetic valence functions

Band structure and DOS calculation

Do it after a converged scf (will use `=.dens`)

```
$ fedit14.00-49-x86_64
```

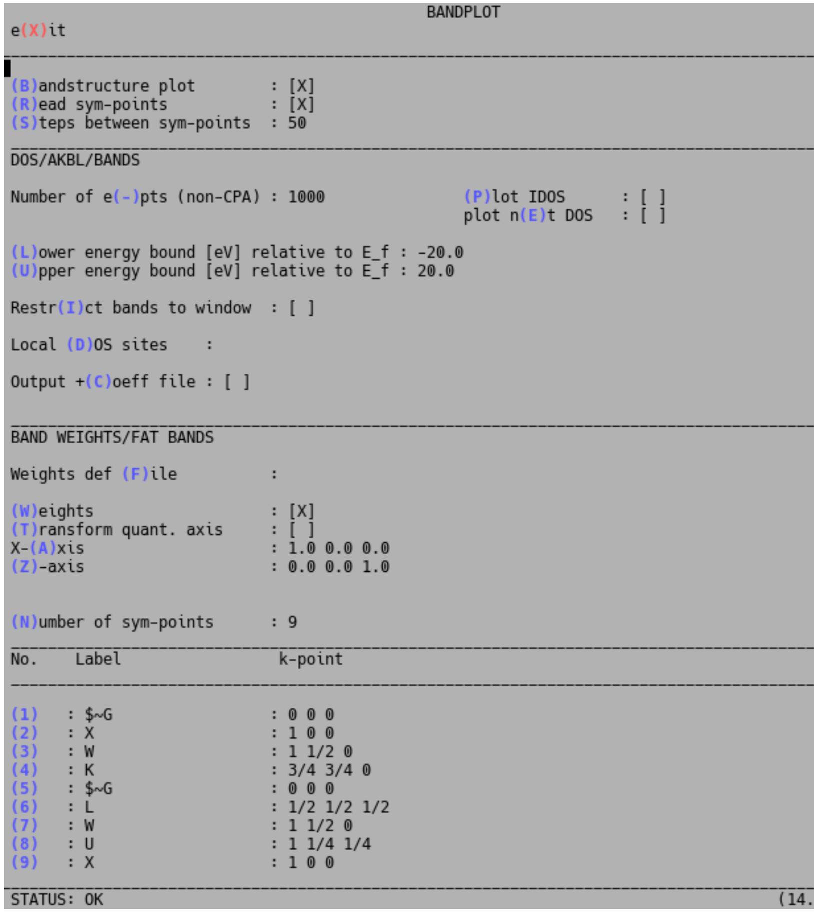
Type `space` `B` to enter Bandplot menu

Type `B` to compute bands

Type `W` to compute weights

Type `X` `Q` `y` to quit

```
$ fplo14.00-49-x86_64 | tee out.band
```



```
                                                    BANDPLOT
e(X)it
--------------------------------------------------------------------------------------------
(B)andstructure plot         : [X]
(R)ead sym-points            : [X]
(S)teps between sym-points   : 50
--------------------------------------------------------------------------------------------
DOS/AKBL/BANDS

Number of e(-)pts (non-CPA) : 1000                 (P)lot IDOS      : [ ]
                                                   plot n(E)t DOS   : [ ]

(L)ower energy bound [eV] relative to E_f : -20.0
(U)pper energy bound [eV] relative to E_f : 20.0

Restr(I)ct bands to window   : [ ]

Local (D)OS sites     :

Output +(C)oeff file : [ ]

--------------------------------------------------------------------------------------------
BAND WEIGHTS/FAT BANDS

Weights def (F)ile           :

(W)eights                    : [X]
(T)ransform quant. axis      : [ ]
X-(A)xis                     : 1.0 0.0 0.0
(Z)-axis                     : 0.0 0.0 1.0


(N)umber of sym-points       : 9
--------------------------------------------------------------------------------------------
No.    Label                  k-point
--------------------------------------------------------------------------------------------

(1)   : $~G                  : 0 0 0
(2)   : X                    : 1 0 0
(3)   : W                    : 1 1/2 0
(4)   : K                    : 3/4 3/4 0
(5)   : $~G                  : 0 0 0
(6)   : L                    : 1/2 1/2 1/2
(7)   : W                    : 1 1/2 0
(8)   : U                    : 1 1/4 1/4
(9)   : X                    : 1 0 0
--------------------------------------------------------------------------------------------
STATUS: OK                                                                                (14.
```

Band structure and DOS calculation

`+band/+bweights` Band structure and band weights

`+dos.sort*` Projected DOS

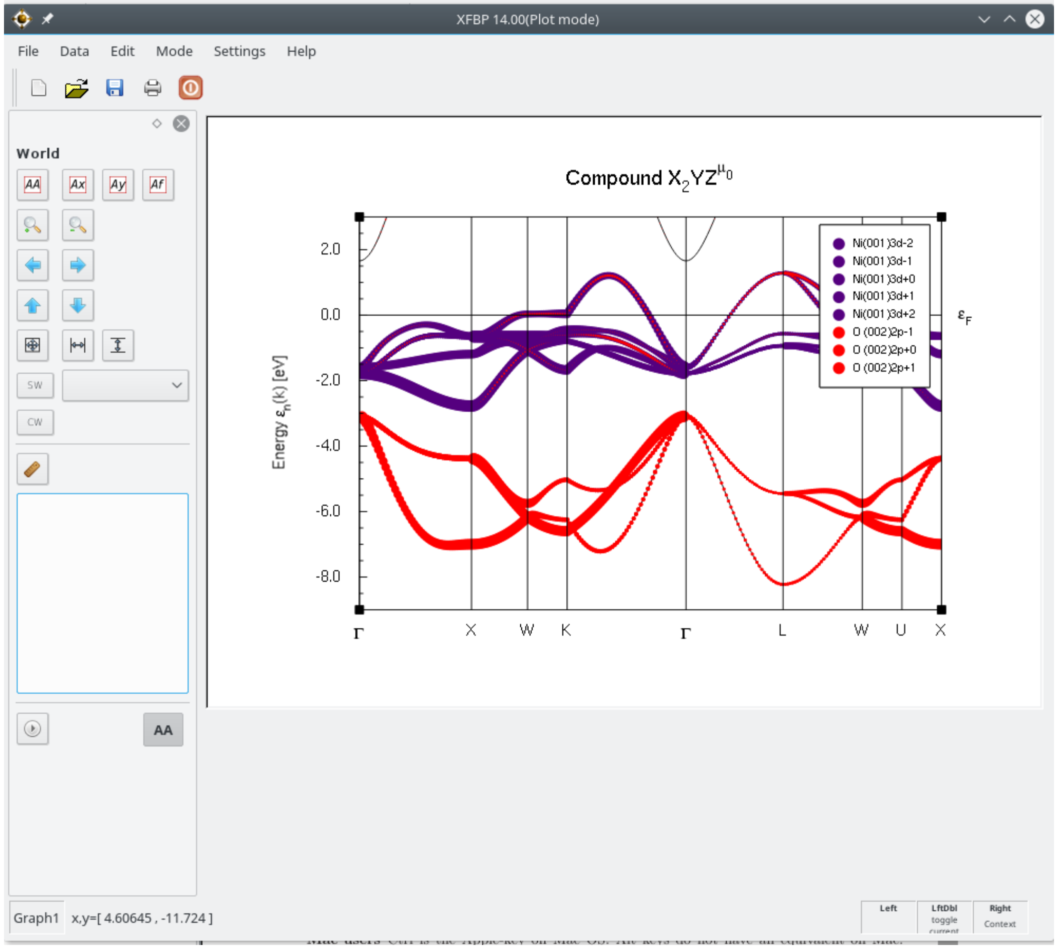
```
$ head -n 1 +dos.sort001.nl00[1-7]
```

```
==> +dos.sort001.nl001 <==
# sort =     1  nl = 3s spin =     1

==> +dos.sort001.nl002 <==
# sort =     1  nl = 3p spin =     1

==> +dos.sort001.nl003 <==
# sort =     1  nl = 4s spin =     1

==> +dos.sort001.nl004 <==
# sort =     1  nl = 5s spin =     1

==> +dos.sort001.nl005 <==
# sort =     1  nl = 3d spin =     1

==> +dos.sort001.nl006 <==
# sort =     1  nl = 4d spin =     1

==> +dos.sort001.nl007 <==
# sort =     1  nl = 4p spin =     1
```

Band structure and DOS plot

FPLO includes a plotting programm: `XFBP`

```
$ xfbp +bweights
```



● Ni 3*d*

● O 2*p*

(weights added by double cliking on the graph)

Non magnetic DFT calculations → NiO described as conductor

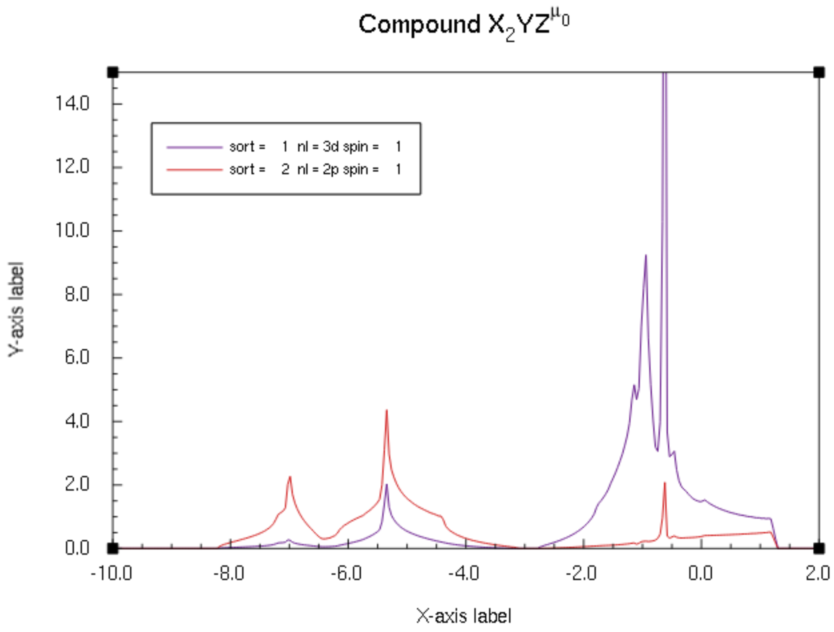
Failing of DFT to describe highly correlated materials

Band structure and DOS plot

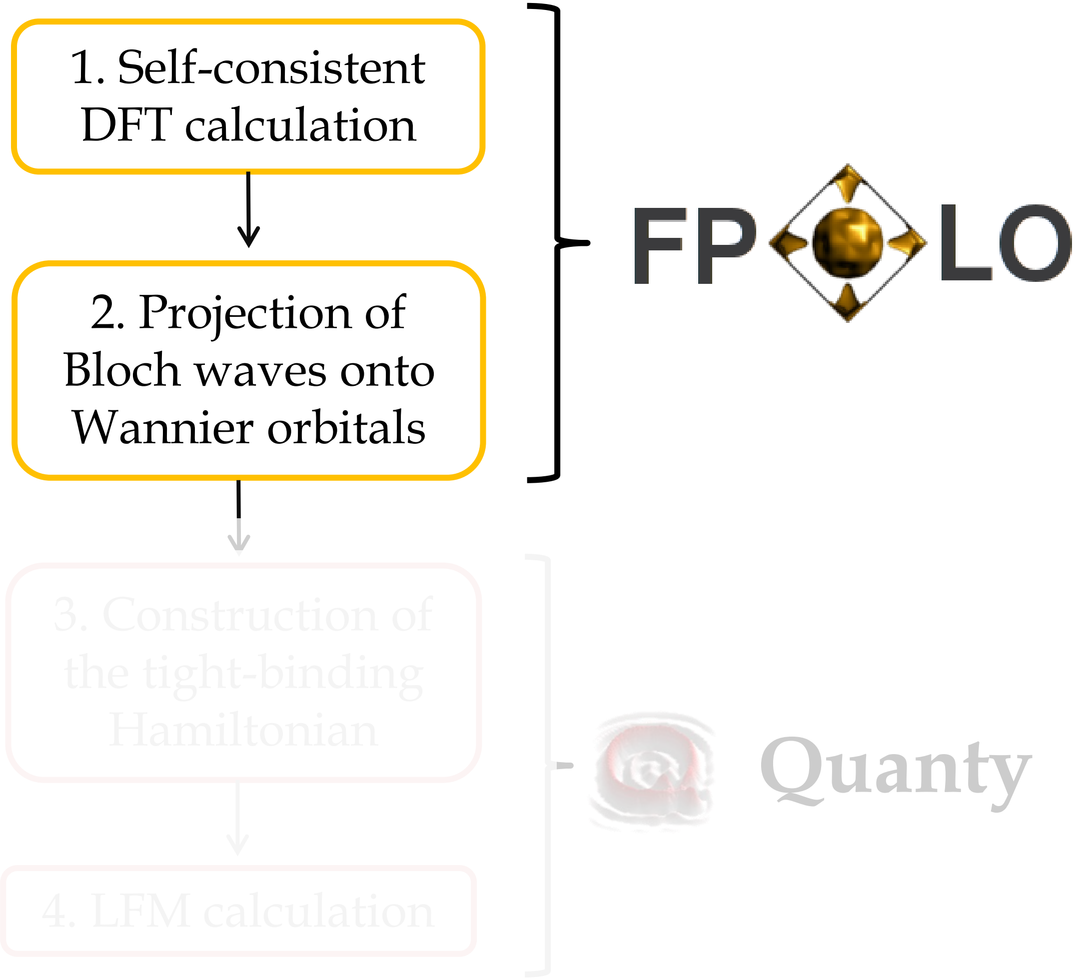
FPLO includes a plotting programm: `XFBP`

```
$ xfbp +dos.sort001.nl005 +dos.sort002.nl004
```

DOS (states/eV) *vs* Energy (eV)



“Ab-initio” Ligand Field Multiplet: procedure



1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

FPLO

3. Construction of the tight-binding Hamiltonian

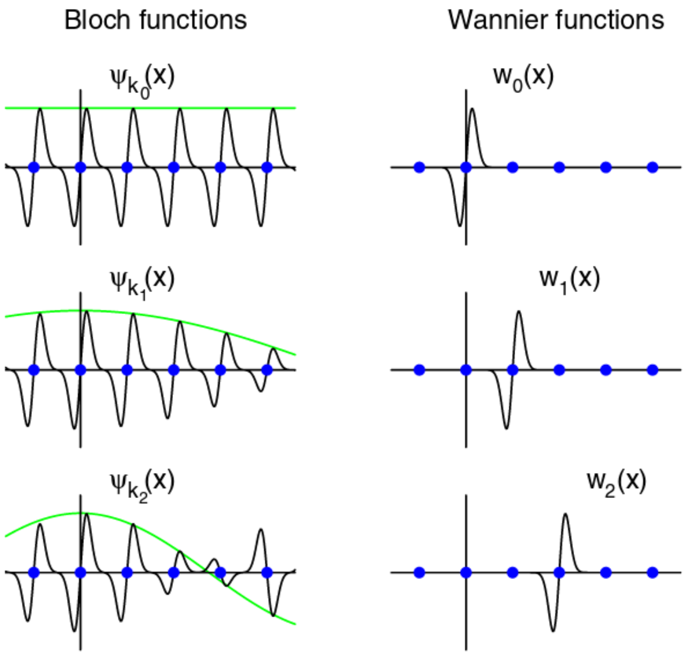
4. LFM calculation

Quanty

Wannier orbitals

Wannier functions:

- set of orthonormal localized functions
- span the same space as the Bloch functions



Marzari et *al*. Rev. Mod. Phys., Vol. 84, No. 4

$$W_\mu(\mathbf{r}-\boldsymbol{T}) = \frac{V}{(2\Pi)^3}\int_{BZ} \mathrm{d}\boldsymbol{k}\, e^{-i\mathbf{k}.\mathbf{T}} \sum_n \psi_n^{\boldsymbol{k}}(\boldsymbol{r})\, U^{\boldsymbol{k}}_{n\,\mu}$$

Kohn-Sham functions

$U^{\boldsymbol{k}}$: Unitary matrix

Projections onto Wannier functions can be interfaced with any band structure method

FPLO allows to obtain highly localized atom-like Wannier functions (≠ maximally localized)

Wannier functions with FPLO

The Wannier module of FPLO is activated if:

- A file `=.wandef` is present in the folder
- The keyword `doit` is found in the file

```
$ cp ../=.wandef
```

```
$ fplo14.00-49-x86_64 | tee out.wandef
```

Compute the projection matrix (still in *k* space)

```
$ fplo14.00-49-x86_64 | tee out.wan
```

Compute Wannier functions

Wannier functions with FPLO

File = `.wandef`

```
doit

---- real space grid for pictures of WFs ---------------------

WF_grid_basis conv
WF_grid_directions
2 0 0
0 2 0
0 0 2
WF_grid_subdivision  30 30 30
-------------------------------------------------
```

Output of the WFs on the real space grid (visualization)

Real space grid size. You can set it to 1 1 1 if you do not need to plot the orbitals (save time and memory)

```
ham_cutoff 18
WF_ham_threshold 0
WF_coeff_threshold 0
WF_write_coeff_stats on
ham_write_t_stats on
```

In the output only hopping elements |t|>`WF_ham_threshold` are written, for atoms distances less than `ham_cutoff`.

Only contributions of the FPLO orbitals to the WFs, which are larger than `WF_coeff_threshold` are written in `+WF_coefficients`

```
----- ham export grid --------

k_grid_basis prim
k_grid_directions
1 0 0
0 1 0
0 0 1
k_grid_subdivision  1 1 1
k_grid_incl_periodic_points off
```

Output of the WFs on the reciprocal space grid

Wannier functions with FPLO

File = `.wandef`

One WF projector

```
-------- NiO --------------

wandef
on
  name Ni 3dxy
  emin -9
  emax   +1.5
  de 0.0
  contrib
    site 1
    difvec 0 0 0
    xaxis 1 0 0
    zaxis 0 0 1
    orb 3d-2
    fac    1
wandef
on
  name Ni 3dyz
  emin -9
  emax   +1.5
  de 0.0
  contrib
    site 1
    difvec 0 0 0
    xaxis 1 0 0
    zaxis 0 0 1
    orb 3d-1
    fac    1
…
```

Energy window: 1 from `emin` to `emax`, falling like a gaussian of width `de`

Orbital sitting at site 1

`difvec` must be non zero if several contributions in the Wannier function

`xaxis zaxis`: local coordinate system

Orbital 3d-2 →

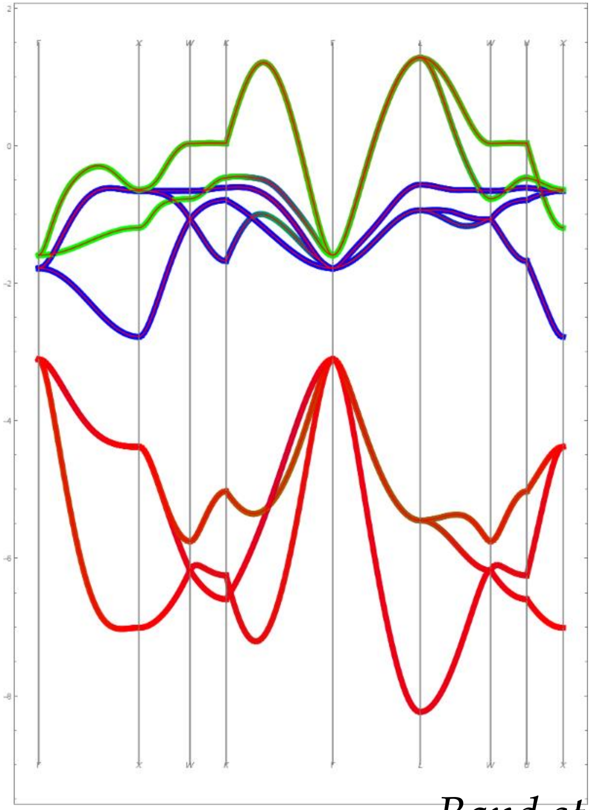
`fac` is the relative weight of each contrib

-2	-1	0	1	2
xy	yz	z^2	xz	x^2-y^2

There can be more sites than Wyckoff postions (check scf output)
Define Wannier orbitals for each of them

Check band structure and plot orbitals

If you have mathematica open file **`NiO_wannier.nb`** in folder **`NiO`**



Band structure calculated
(i) with the full basis set (red lines) and
(ii) with the Wannier-functions basis set.

Green: Ni 3*d* e_g
Blue: Ni 3*d* $t2_g$
Red: O 2*p*

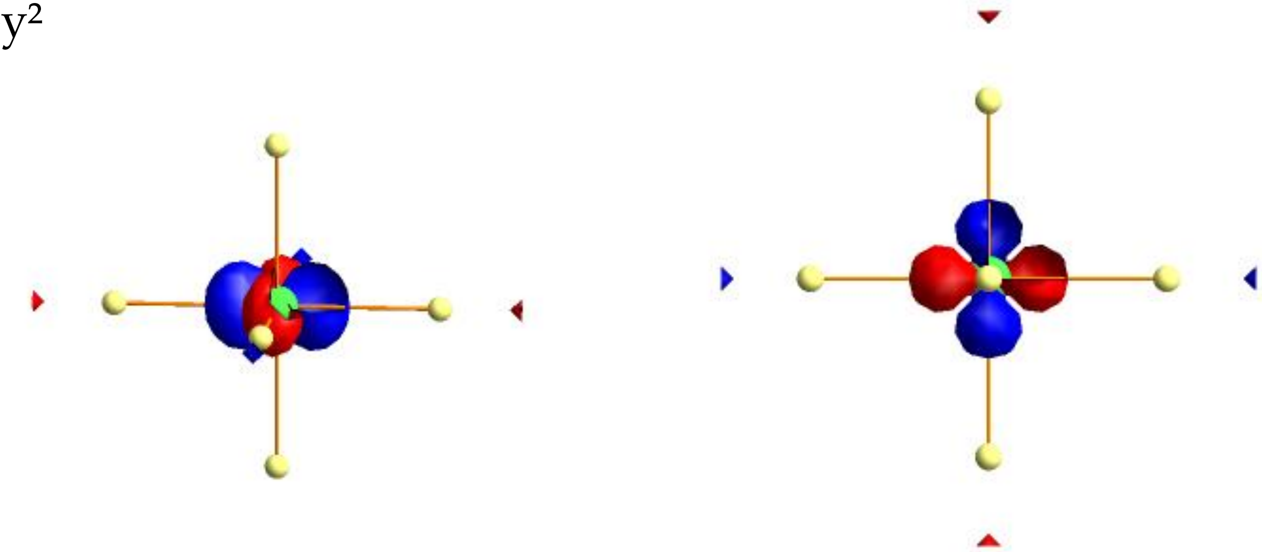
→ superimposed

Band structure plot will be implemented in Quanty in a near future

Check band structure and plot orbitals

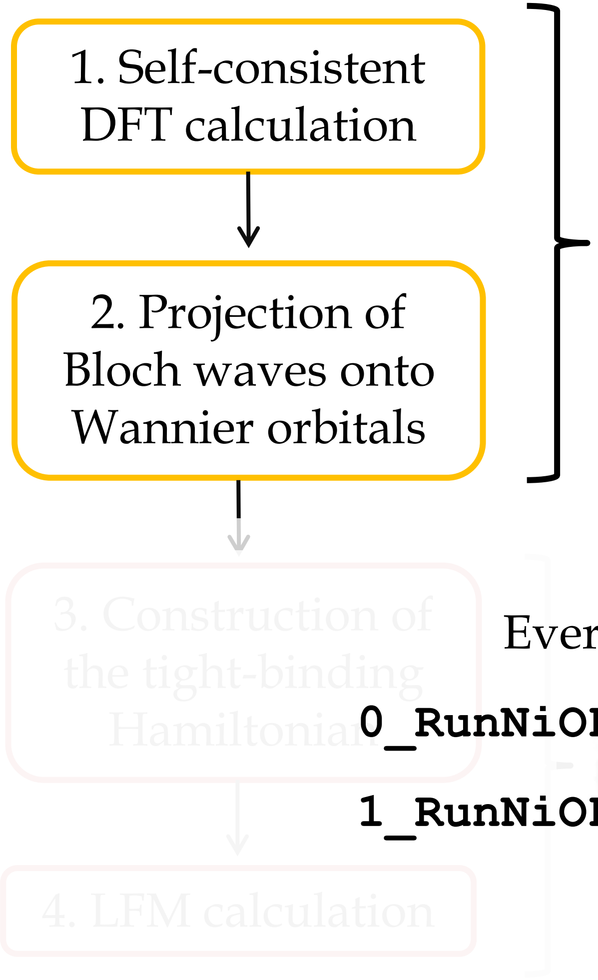
If you have mathematica open file **`NiO_wannier.nb`** in folder **`NiO`**

Wannier orbitals:
e.g. Ni 3*d* x²-y²



Script method

```
$ cd Scripts
```



1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

3. Construction of the tight-binding Hamiltonian

4. LFM calculation

(i) Scf with low convergence criteria
(ii) Better scf calculation
(iii) Band structure calculation

(iv) Projection onto Wannier orbitals

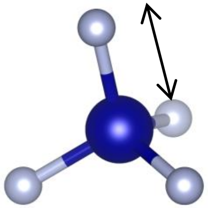
Everything in two scripts:

`0_RunNiOFPLO.Quanty` ((i) to (iii))

`1_RunNiOFPLO_WF.Quanty` (iv)

Scripts could be written in any language (bash, python...)

Example II: molecule CrF_4



Unknown bond length → needs to be optimized

Procedure:

(i) Optimization of atomic positions
(ii) Well converged scf calculation
(iii) Band structure calculation
(iv) Projection onto Wannier orbitals

Use the script for NiO as a starting point:

```
$   cp NiO/Scripts/0_RunNiOFPLO.Quanty CrF4
$   cd CrF4
$   mv 0_RunNiOFPLO.Quanty 0_RunCrF4FPLO.Quanty
```

} Rename file

Example II: molecule CrF_4

- Modify file `0_RunCrF4FPLO.Quanty`

```
# title
@c@CrF4
# enter spacegroup select box - 215
@s@
@215@
@x@
# structure type - molecule
@t@
@m@
@x@
```

~~`# lattice constants;`~~
~~`@l@ 4.178   4.178   4.178`~~
~~`# set axis angles`~~
~~`@a@ 90. 90. 90.`~~

Not needed for a molecule

```
# setup Wyckoff positions
@n@2
# Now, give list of all Wyckoff positions.
@1@ Cr @ 0. 0. 0.
@2@ F @ 1. 1. 1.
```

Example II: molecule CrF_4

- Modify file `0_RunCrF4FPLO.Quanty`

In the first FEditMenuOptions (below the structure)

```
# Set maximum number of iterations to 200
@n@200
# Set convergence criterion to Density OR energy
@g@
@1@
@x@
# activate structure optimization
# (The space before the 'f' opens the alternative menu bar.)
@ f@
@f@
@2@
@x@
# leave forces menu
@x@
```

Lines to be added for structural optimization

Example II: molecule CrF_4

- Modify file `0_RunCrF4FPLO.Quanty`

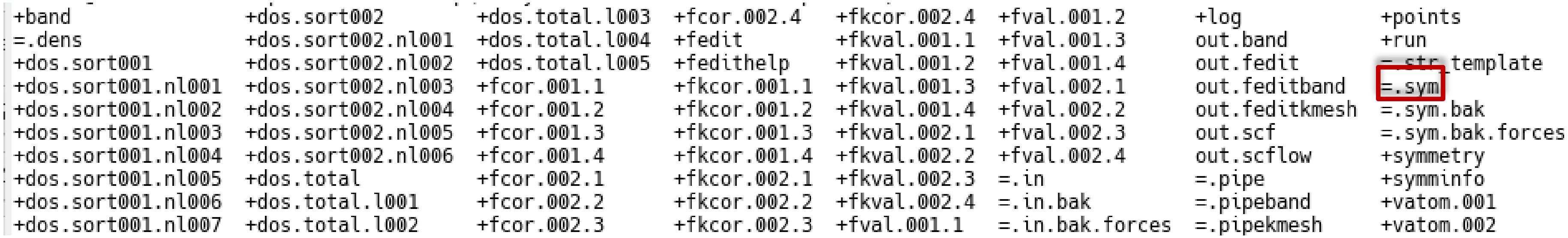
In the second FEditMenuOptions

```
# deactivate site geometry optimization again
# (The space before the 'f' opens the alternative menu bar.)
@ f@
@f@
@1@
@x@
# leave forces menu
@x@
```

- `$  Quanty 0_RunCrF4FPLO.Quanty`

Example II: molecule CrF_4

```
$   cd DFT
$   ls
```



```
+band               +dos.sort002        +dos.total.l003  +fcor.002.4   +fkcor.002.4  +fval.001.2      +log            +points
=.dens              +dos.sort002.nl001  +dos.total.l004  +fedit        +fkval.001.1  +fval.001.3      out.band        +run
+dos.sort001        +dos.sort002.nl002  +dos.total.l005  +fedithelp    +fkval.001.2  +fval.001.4      out.fedit       =.str_template
+dos.sort001.nl001  +dos.sort002.nl003  +fcor.001.1      +fkcor.001.1  +fkval.001.3  +fval.002.1      out.feditband   =.sym
+dos.sort001.nl002  +dos.sort002.nl004  +fcor.001.2      +fkcor.001.2  +fkval.001.4  +fval.002.2      out.feditkmesh  =.sym.bak
+dos.sort001.nl003  +dos.sort002.nl005  +fcor.001.3      +fkcor.001.3  +fkval.002.1  +fval.002.3      out.scf         =.sym.bak.forces
+dos.sort001.nl004  +dos.sort002.nl006  +fcor.001.4      +fkcor.001.4  +fkval.002.2  +fval.002.4      out.scflow      +symmetry
+dos.sort001.nl005  +dos.total          +fcor.002.1      +fkcor.002.1  +fkval.002.3  =.in             =.pipe          +symminfo
+dos.sort001.nl006  +dos.total.l001     +fcor.002.2      +fkcor.002.2  +fkval.002.4  =.in.bak         =.pipeband      +vatom.001
+dos.sort001.nl007  +dos.total.l002     +fcor.002.3      +fkcor.002.3  +fval.001.1   =.in.bak.forces  =.pipekmesh     +vatom.002
```

=.sym contains information on the optimized structure

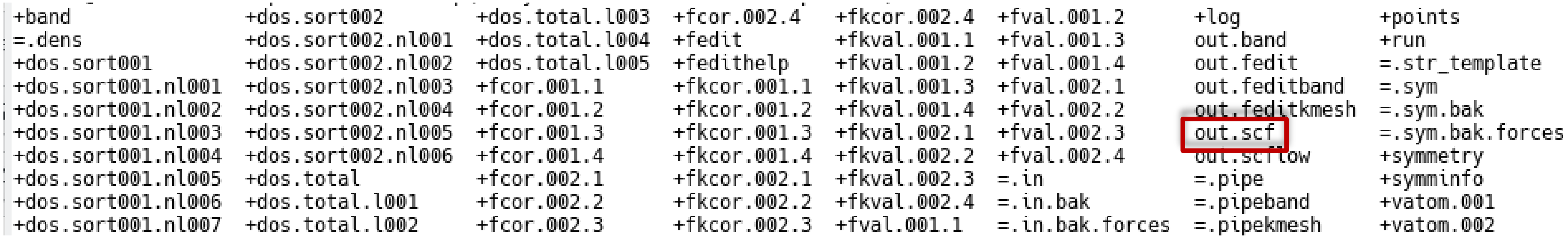
Do not manipulate this file (written by fedit)

Optimized position of F

```
F: 0.9826012122  0.9826012122   0.9826012122
```

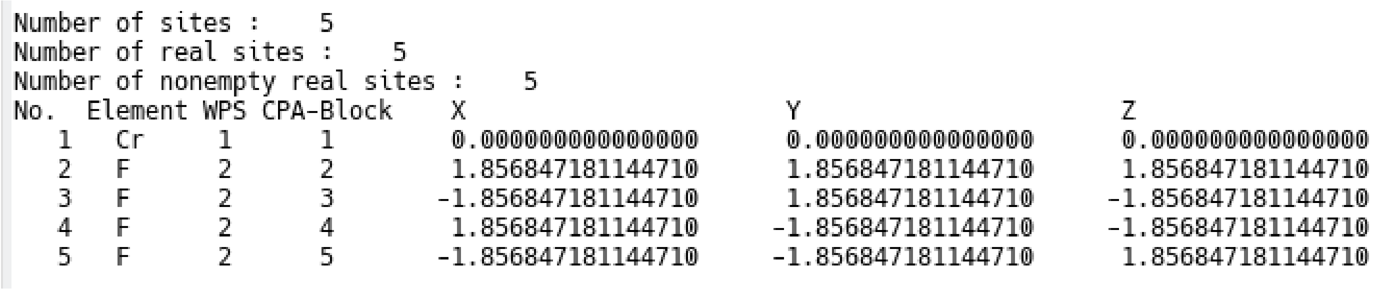
Example II: molecule CrF_4

```
$   cd DFT
$   ls
```



```
+band               +dos.sort002        +dos.total.l003  +fcor.002.4   +fkcor.002.4  +fval.001.2      +log            +points
=.dens              +dos.sort002.nl001  +dos.total.l004  +fedit        +fkval.001.1  +fval.001.3      out.band        +run
+dos.sort001        +dos.sort002.nl002  +dos.total.l005  +fedithelp    +fkval.001.2  +fval.001.4      out.fedit       =.str_template
+dos.sort001.nl001  +dos.sort002.nl003  +fcor.001.1      +fkcor.001.1  +fkval.001.3  +fval.002.1      out.feditband   =.sym
+dos.sort001.nl002  +dos.sort002.nl004  +fcor.001.2      +fkcor.001.2  +fkval.001.4  +fval.002.2      out.feditkmesh  =.sym.bak
+dos.sort001.nl003  +dos.sort002.nl005  +fcor.001.3      +fkcor.001.3  +fkval.002.1  +fval.002.3      out.scf         =.sym.bak.forces
+dos.sort001.nl004  +dos.sort002.nl006  +fcor.001.4      +fkcor.001.4  +fkval.002.2  +fval.002.4      out.scflow      +symmetry
+dos.sort001.nl005  +dos.total          +fcor.002.1      +fkcor.002.1  +fkval.002.3  =.in             =.pipe          +symminfo
+dos.sort001.nl006  +dos.total.l001     +fcor.002.2      +fkcor.002.2  +fkval.002.4  =.in.bak         =.pipeband      +vatom.001
+dos.sort001.nl007  +dos.total.l002     +fcor.002.3      +fkcor.002.3  +fval.001.1   =.in.bak.forces  =.pipekmesh     +vatom.002
```

In **out.scf**, the sites are listed



```
Number of sites :     5
Number of real sites :     5
Number of nonempty real sites :     5
No.  Element WPS CPA-Block     X                     Y                     Z
    1    Cr      1       1       0.0000000000000000    0.0000000000000000    0.0000000000000000
    2    F       2       2       1.856847181144710     1.856847181144710     1.856847181144710
    3    F       2       3      -1.856847181144710     1.856847181144710    -1.856847181144710
    4    F       2       4       1.856847181144710    -1.856847181144710    -1.856847181144710
    5    F       2       5      -1.856847181144710    -1.856847181144710     1.856847181144710
```

The 4 F sites must be listed in the =.wandef files,
even if they correspond to the same Wyckoff position

Example II: molecule CrF_4

- Have a look at the file `1_RunCrF4FPLO_WF.Quanty`

```
WanDef=[[
…
wandef
on
    name Cr_3d-2
    emin -10
    emax   10
    de 0
    contrib
       site 1
       difvec 0 0 0
       xaxis 1 0 0
       zaxis 0 0 1
       orb 3d-2
       fac    1

…]]
```

For Cr atom 5 similar definitions:
3d-2   3d-1   3d0   3d1   3d2

Example II: molecule CrF_4

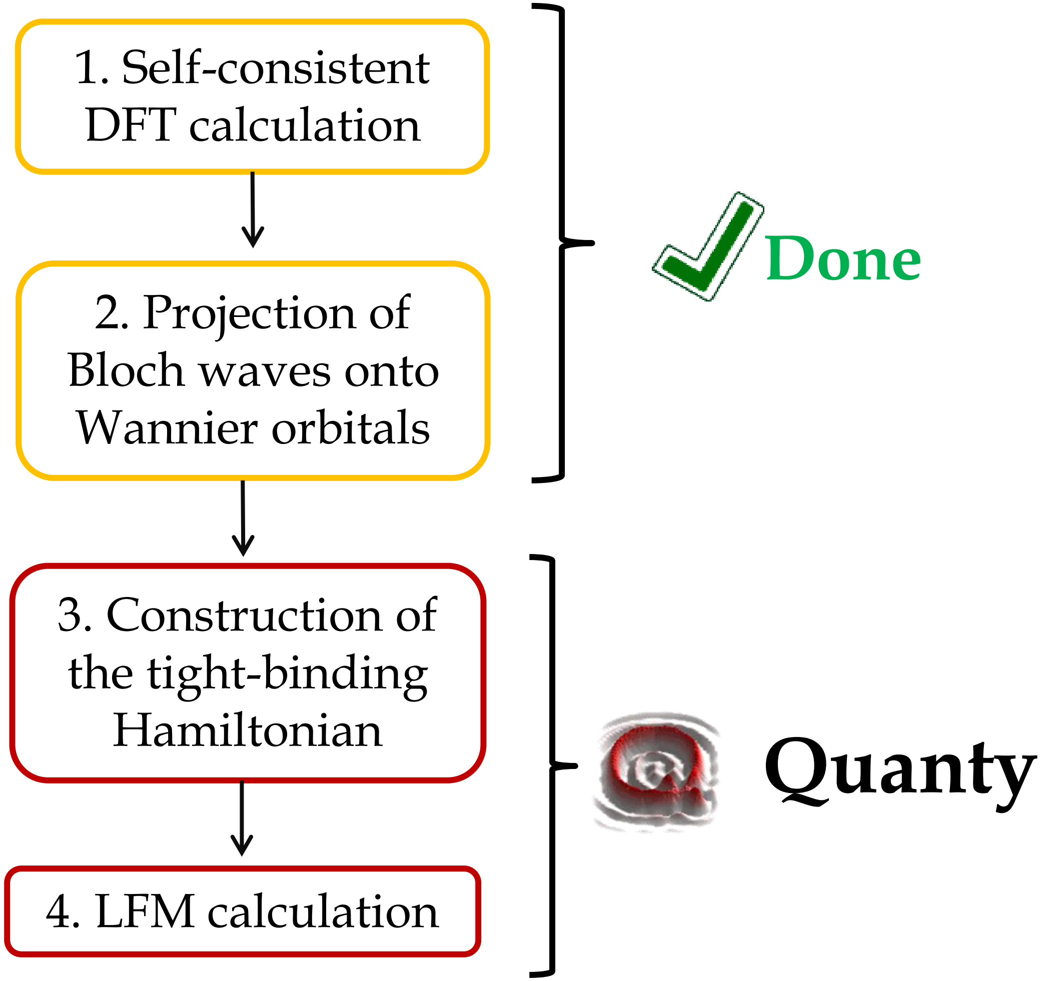
- Have a look at the file `1_RunCrF4FPLO_WF.Quanty`

For F atoms, the script can include a loop over the sites to avoid writing 4 times the same thing:

```
for i=1,4 do
WanDef = WanDef..[[

wandef
on    name F_]]..i..[[_3p-1
    emin -10
    emax   10
    de 0
    contrib
       site ]]..(i+1)..[[
       difvec 0 0 0
       xaxis 1 0 0
       zaxis 0 0 1
       orb 3p-1
       fac    1
…]]
```

- **`$   Quanty 0_RunCrF4FPLO.Quanty`**

Script method



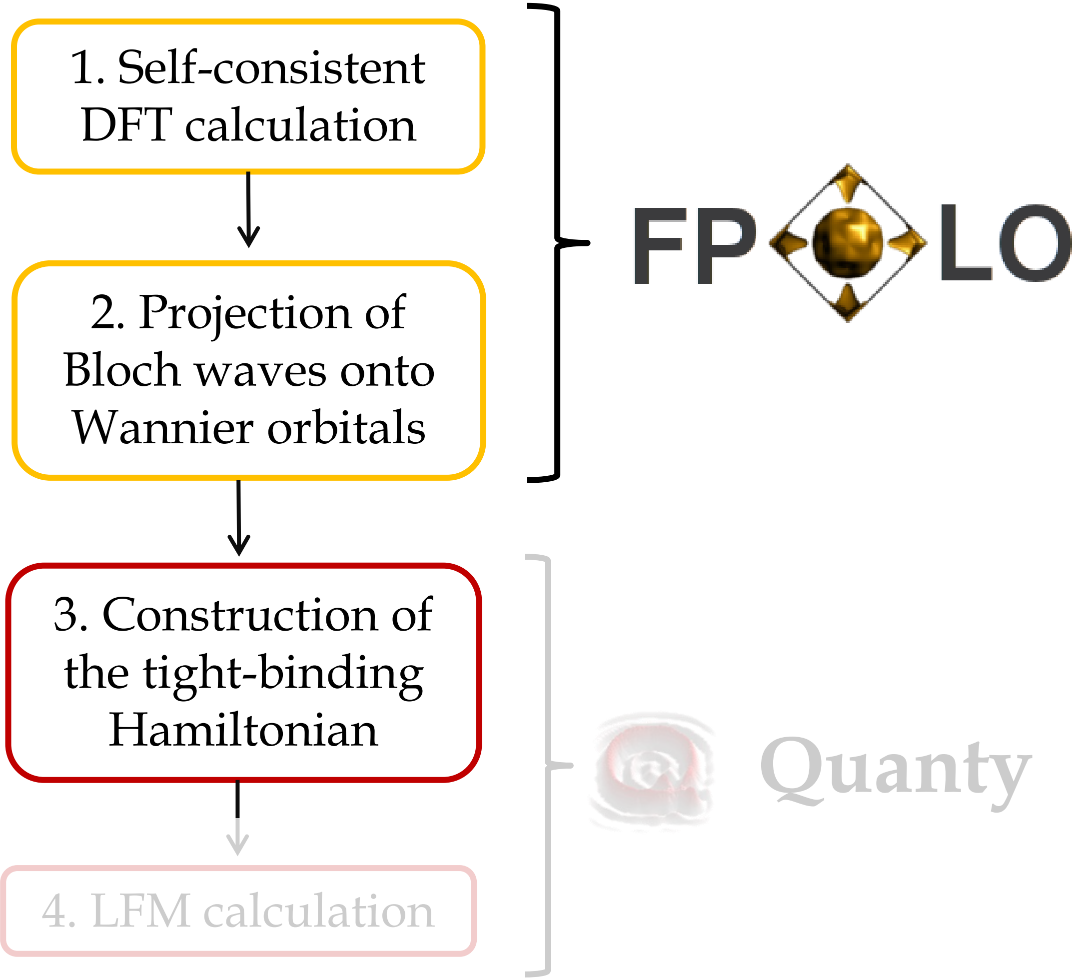
1. Self-consistent DFT calculation
2. Projection of Bloch waves onto Wannier orbitals
3. Construction of the tight-binding Hamiltonian
4. LFM calculation

✓ **Done**

Quanty

Part 2: Tight binding Hamiltonian and LFM calculation

“*Ab-initio*” Ligand Field Multiplet: procedure



1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

3. Construction of the tight-binding Hamiltonian

4. LFM calculation

FPLO

Quanty

Example I: NiO - Ground state calculation

View the script **`2_groundstate.Quanty`** in folder **`NiO/LFMcalc`**

Tight-Binding Hamiltonian

- Read the file out.wan and

```
FPLOOut = FileReadDresdenFPLO("DFT/out.wan")
```

- Create the tight binding Hamiltonian for the crystal

```
TB = TightBindingDefFromDresdenFPLO(FPLOOut)
```

```
print(TB)
```

Display all the hopping parameters in the crystal

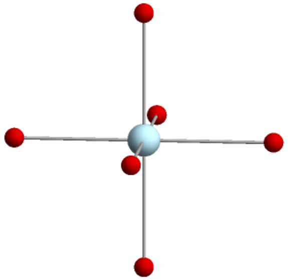
Tight-Binding Hamiltonian

- Choice of a cluster of atoms

Find all atoms inside a sphere of radius 4 centered at the origin:

```
NewCluster =
FindAllAtomsInsideSphere(TB.Atoms,TB.Cell,{0,0,0},4)
```

Ni surrounded by 6 O
3.95 Bohr away



=> 5 Ni *d* and 6*3 O *p*

- Construction of the matrix H_{TB} $(H_{TB})_{\mu'\mu} = \langle W_{\mu'} | H_{\mathrm{DFT}} | W_{\mu} \rangle$

```
HDFTLarge, ClusterTB = CreateClusterHamiltonian(TB, {"open",
NewCluster},  {{"AddSpin",true}})
```

```
print(HDFTLarge)
print(OperatorToMatrix(Hamiltonian))
```

Operator form

Matrix form
(23x23 matrix)

$$H = \left(\begin{array}{c|c} 3d \text{ orbitals} & 3d - L \text{ coupling} \\ \hline 3d - L \text{ coupling} & L \text{ orbitals} \end{array} \right)$$

Linear combination of ligand orbitals

- Compute rotation matrix

Some linear combinations of ligand p orbitals do not interact with the metal *d* orbitals ⇒Possibly drastic reduction of the basis set size

In *Oh* symmetry 5 ligand orbitals (two e_g and three t_{2g}) instead of 18 orbitals

Symmetry Adapted Linear Combination (see Ballhausen - *Introduction fo Ligand Field Theory* - p.154)

$$
\begin{array}{cccccccccccccccccc}
0. & 0. & 0. & 0. & 0. & 0.5 & -0.5 & 0. & 0. & 0.5 & 0. & 0. & 0. & 0. & -0.5 & 0. & 0. & 0. \\
0.5 & 0. & 0. & 0. & 0.5 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & -0.5 & 0. & -0.5 & 0. & 0. \\
0. & -0.58 & 0. & 0.29 & 0. & 0. & 0. & 0. & -0.29 & 0. & 0. & 0.29 & -0.29 & 0. & 0. & 0. & 0.58 & 0. \\
0. & 0. & 0.5 & 0. & 0. & 0. & 0. & -0.5 & 0. & 0. & 0.5 & 0. & 0. & 0. & 0. & 0. & 0. & -0.5 \\
0. & 0. & 0. & 0.5 & 0. & 0. & 0. & 0. & 0.5 & 0. & 0. & -0.5 & -0.5 & 0. & 0. & 0. & 0. & 0.
\end{array}
$$

In Quanty, the block band diagonalize algorithm computes the required rotation matrix in general symmetry:

```
RTB, T = BlockBandDiagonalize(ClusterTB,{{"Ni",{0,0,0}}})
```

Tight binding object

Rotation matrix

Tight binding object

Starting block
(Ni *d* block will remain unchanged)

```
Print(T)
```

Linear combination of ligand orbitals

- Construction of the rotated tight-binding Hamiltonian

```
HDFT, ClusterTB, IndexFPLO = CreateClusterHamiltonian(RTB,
{"open", RTB.Atoms},  {{"AddSpin",true},
{"ReturnTBIndicesDict",true}})
```

```
print(HDFT)
```

```
print(OperatorToMatrix(HDFT))
```

20x20 matrix (with spin)

The number of orbitals was reduced from 46 to 20

⇒ drastic reduction of the number of possible states from $\binom{46}{44}$=1035 to $\binom{20}{18}$=190

Linear combination of ligand orbitals

- Construction of the rotated tight-binding Hamiltonian

```
HDFT, ClusterTB, IndexFPLO = CreateClusterHamiltonian(RTB,
{"open", RTB.Atoms},  {{"AddSpin",true},
{"ReturnTBIndicesDict",true}})
```

`print(IndexFPLO)` List the automatically generated indices

- Generate more convenient indices.

```
Index, NFermi =
CreateAtomicIndicesDict({"Ni_3d","Ligand_d"})
```

`print(Index)` List the groups of indices

e.g. Ni_3d_Up: 3d orbitals of Ni with spin up
use `Index["Ni_3d_Up"]` to refer to those

“*Ab-initio*” Ligand Field Multiplet: procedure

1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

FPLO

3. Construction of the tight-binding Hamiltonian

4. LFM calculation

Quanty

Hamiltonian

```
Hamiltonian = HDFT
             -F0dd*OppF0MFDFT-F2dd*OppF2MFDFT-F4dd*OppF4MFDFT
             + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
             + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots
```

Hamiltonian

```
Hamiltonian = HDFT
              -F0dd*OppF0MFDFT-F2dd*OppF2MFDFT-F4dd*OppF4MFDFT
              + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
              + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots
```

Coulomb operator: `F0dd*OppF0_3d + F2dd*OppF2_3d + F4dd*OppF4_3d`

```
OppF0 =
Rotate(NewOperator("U",10,Index["Ni_3d_Up"],Index["Ni_3d_Dn"],{1,0,0}),YtoZMatrix("d"))
OppF2 =
Rotate(NewOperator("U",10,Index["Ni_3d_Up"],Index["Ni_3d_Dn"],{0,1,0}),YtoZMatrix("d"))
OppF4 =
Rotate(NewOperator("U",10,Index["Ni_3d_Up"],Index["Ni_3d_Dn"],{0,0,1}),YtoZMatrix("d"))
```

Operators are rotated
to a basis of tesseral harmonics

Within Ni *d* shell

F0,F2 or F4

Hamiltonian

```
Hamiltonian = HDFT
             -F0dd*OppF0MFDFT-F2dd*OppF2MFDFT-F4dd*OppF4MFDFT
             + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
             + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots
```

Coulomb operator: `F0dd*OppF0_3d + F2dd*OppF2_3d + F4dd*OppF4_3d`

Determined from the **parameter Udd** which refers to the center the multiplet

Udd = F0dd+(F2dd+F4dd)*2/63
⇒ F0dd = Udd +(F2dd+F4dd)*2/63

Computed from the DFT radial wave-functions

```
slaterIntegrals = GetSlaterIntegrals({"3d"},correlatedRadialFunctions)
F2dd = slaterIntegrals["3d 3d 3d 3d"][2] * EnergyUnits.Ha.value
F4dd = slaterIntegrals["3d 3d 3d 3d"][4] * EnergyUnits.Ha.value
```

Hamiltonian

```
Hamiltonian = HDFT
            -F0dd*OppF0MFDFT-F2dd*OppF2MFDFT-F4dd*OppF4MFDFT
             + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
             + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots
```

Double counting corrections
Coulomb interactions are included in the Kohn-Sham potential. Addition of the full Coulomb interaction leads to a "double counting" problem. A correction is needed:

(i) Determine DFT density matrix of the *d* shell (5x5 matrix)

(ii) Calculate the Coulomb operator **within mean-field approximation** including self interaction (in KS-DFT the electron interacts with the entire electron density which includes its own density):

```
OppF0MFDFT =
MeanFieldOperator(OppF0,rhod,{{"AddDFTSelfInteraction",true}})
OppF2MFDFT =
MeanFieldOperator(OppF2,rhod,{{"AddDFTSelfInteraction",true}}
OppF4MFDFT =
MeanFieldOperator(OppF4,rhod,{{"AddDFTSelfInteraction",true}})
```

Hamiltonian

```
Hamiltonian = HDFT
             -F0dd*OppF0MFDFT-F2dd*OppF2MFDFT-F4dd*OppF4MFDFT
             + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
             + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots
```

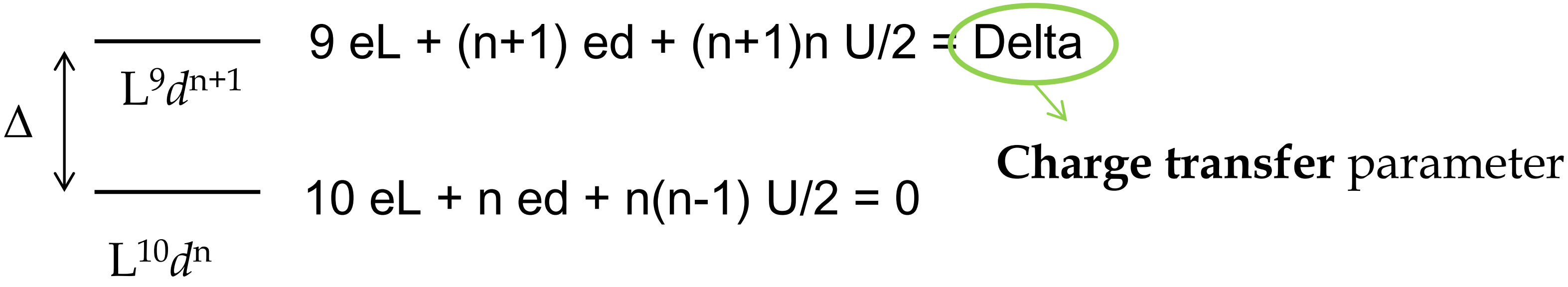
External magnetic field and spin orbit coupling within the Ni 3*d* shell

You can add an exchange field `Hz * OppSz`

Hamiltonian

```
Hamiltonian = HDFT
             -F0dd*OppF0MFDFT-F2dd*OppF2MFDFT-F4dd*OppF4MFDFT
             + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
             + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots
```

- **Set on-site energies:**



9 eL + (n+1) ed + (n+1)n U/2 = Delta

Charge transfer parameter

10 eL + n ed + n(n-1) U/2 = 0

```
ed        =  (10*Delta-nd*(19+nd)*Udd/2)/(10+nd)
eL        = nd*((1+nd)*Udd/2-Delta)/(10+nd)

OperatorSetTrace(Hamiltonian,ed,Index["Ni_3d"])
OperatorSetTrace(Hamiltonian,eL,Index["Ligand_d"])
```

Compute eigenvectors

- Compute eigenvectors

```
StartRestrictions = {NFermi, 0,
{DeterminantString(NFermi,Index["Ni_3d"]),nd,nd},
{DeterminantString(NFermi,Index["Ligand_d"]),10,10}}
```

```
psiList = Eigensystem(Hamiltonian, StartRestrictions, Npsi)
```

```
$  Quanty 2_groundstate.Quanty
```

⇒ Print the energy and expectation values of several operators
for the `Npsi` first eigenstates

Example I: NiO - XAS Calculation

```
$  cp 2_groundstate.Quanty 3_XAScalc.Quanty
```

and modify **`3_XAScalc.Quanty`**

Or copy **`3_XAS.Quanty`** from ../Scripts

- Add the Ni 2*p* shell in the index list

Change line:

```
Index, NFermi =
CreateAtomicIndicesDict({"Ni_3d","Ligand_d","Ni_2p"},
{{"Ni",{"Ni_2p","Ni_3d"}}})
```

Add line:

```
HDFT.NF = NFermi
```

The number of fermionic states is now 26

Example I: NiO - XAS Calculation

- Define the 2*p*-3*d* Coulomb repulsion operator

Add lines:

```
OppUpdF0 = Rotate(NewOperator("U", NFermi,
Index["Ni_2p_Up"],Index["Ni_2p_Dn"],
Index["Ni_3d_Up"],Index["Ni_3d_Dn"], {1,0},
{0,0}),YtoZMatrix({"Ni_3d","Ligand_d","Ni_2p"}))
OppUpdF2 = Rotate(NewOperator("U", NFermi,
Index["Ni_2p_Up"],Index["Ni_2p_Dn"],
Index["Ni_3d_Up"],Index["Ni_3d_Dn"], {0,1},
{0,0}),YtoZMatrix({"Ni_3d","Ligand_d","Ni_2p"}))
OppUpdG1 = Rotate(NewOperator("U", NFermi,
Index["Ni_2p_Up"],Index["Ni_2p_Dn"],
Index["Ni_3d_Up"],Index["Ni_3d_Dn"], {0,0},
{1,0}),YtoZMatrix({"Ni_3d","Ligand_d","Ni_2p"}))
OppUpdG3 = Rotate(NewOperator("U", NFermi,
Index["Ni_2p_Up"],Index["Ni_2p_Dn"],
Index["Ni_3d_Up"],Index["Ni_3d_Dn"], {0,0},
{0,1}),YtoZMatrix({"Ni_3d","Ligand_d","Ni_2p"}))
```

Example I: NiO - XAS Calculation

- Compute 2*p*-3*d* Slater integrals

Add line:

```
Upd      =  8.5
```

Add line:

```
correlatedRadialFunctionsp =
ReadFPLOBasisFunctions({"2p"},"DFT/+fcor.001.1")
```

Change line:

```
slaterIntegrals =
GetSlaterIntegrals({"2p","3d"},correlatedRadialFunctions)
```

Add lines:

```
F2pd     =  slaterIntegrals["3d 2p 3d 2p"][2] *
EnergyUnits.Ha.value
G1pd     =  slaterIntegrals["2p 3d 3d 2p"][1] *
EnergyUnits.Ha.value
G3pd     =  slaterIntegrals["2p 3d 3d 2p"][3] *
EnergyUnits.Ha.value
F0pd     = Upd + (1/15)*G1pd + (3/70)*G3pd
```

Example I: NiO - XAS Calculation

- Construct a larger rotation matrix

Change line:

~~`YtoZdouble = YtoZMatrix({"Ni_3d","Ligand_d"})`~~

```
YtoZtriple = YtoZMatrix({"Ni_3d","Ligand_d","Ni_2p"})
```

Replace **all** `YtoZdouble` by `YtoZtriple`

- Change the definition of the `OppN_Ni` to include 2*p* states

Change line:

```
NewOperator("Number",NFermi,Index["Ni"],  Index["Ni"],
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1})
```

Example I: NiO - XAS Calculation

- Define electric dipole transition operators for several polarization vectors

Add lines:

```
t=math.sqrt(1/2)

Akm = {{1,-1,t},{1, 1,-t}}
TXASx = NewOperator("CF", NFermi, Index["Ni_3d_Up"],
Index["Ni_3d_Dn"], Index["Ni_2p_Up"], Index["Ni_2p_Dn"],
Akm)
Akm = {{1,-1,t*I},{1, 1,t*I}}
TXASy = NewOperator("CF", NFermi, Index["Ni_3d_Up"],
Index["Ni_3d_Dn"], Index["Ni_2p_Up"], Index["Ni_2p_Dn"],
Akm)
Akm = {{1,0,1}}
TXASz = NewOperator("CF", NFermi, Index["Ni_3d_Up"],
Index["Ni_3d_Dn"], Index["Ni_2p_Up"], Index["Ni_2p_Dn"],
Akm)

TXASr = t*(TXASx - I * TXASy)
TXASl =-t*(TXASx + I * TXASy)
```

Example I: NiO - XAS Calculation

- Define a Hamiltonian including *p*-*d* interactions and 2*p* spin orbit coupling

Add line:

```
zeta_2p = 11.51
```

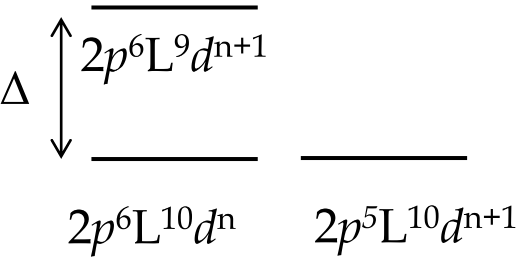
Add lines:

```
XASHamiltonian = HDFT - F0dd * OppF0MFDFT - F2dd *
OppF2MFDFT - F4dd * OppF4MFDFT
      + F0dd * OppF0 + F2dd * OppF2 + F4dd * OppF4
       + Bz * (2*OppSz + OppLz) + zeta_3d * Oppldots_3d
       + zeta_2p * Oppldots_2p
        + F0pd * OppUpdF0 + F2pd * OppUpdF2 + G1pd *
OppUpdG1 + G3pd * OppUpdG3
```

You can add an exchange field `Hz * OppSz`

Example I: NiO - XAS Calculation

- Add on-site energies



6 ep + 9 eL + (n+1) ed + (n+1)n Udd/2 + 6 (n+1) Upd = Delta

5 ep + 10 eL + (n+1) ed + (n+1)(n) Udd/2 + 5 (n+1) Upd = 0
6 ep + 10 eL + n ed + n(n-1) Udd/2 + 6 n Upd = 0

Add lines:

```
epfinal = (10*Delta + (1+nd)*(nd*Udd/2-(10+nd)*Upd)) / (16+nd)
edfinal = (10*Delta - nd*(31+nd)*Udd/2-90*Upd) / (16+nd)
eLfinal = ((1+nd)*(nd*Udd/2+6*Upd) - (6+nd)*Delta) / (16+nd)

OperatorSetTrace(XASHamiltonian,epfinal,Index["Ni_2p"])
OperatorSetTrace(XASHamiltonian,edfinal,Index["Ni_3d"])
OperatorSetTrace(XASHamiltonian,eLfinal,Index["Ligand_d"])
```

Example I: NiO - XAS Calculation

- Compute eigenstates of Hamiltonian

Change line:

```
Npsi = 3
```

Change line:

```
StartRestrictions = {NFermi, 0,
{DeterminantString(NFermi,Index["Ni_3d"]),nd,nd},
{DeterminantString(NFermi,Index["Ligand_d"]),10,10},
{DeterminantString(NFermi,Index["Ni_2p"]),6,6}}
```

Example I: NiO - XAS Calculation

- Compute the spectra for the `Npsi` lowest eigenstates

Add lines:

```
XASSpectra = CreateSpectra(XASHamiltonian, {TXASz, TXASr,
TXASl}, psiList, {{"Emin",-15}, {"Emax",25}, {"NE",2000},
{"Gamma",0.1}})
```

Broadening

```
XASSpectra.Broaden(0.4, {{-3.7, 0.45}, {-2.2, 0.65}, { 0.0,
0.65}, { 1.0, 2.00}, { 6  , 2.00}, { 8  , 0.80}, {13.2,
0.80}, {14.0, 0.90}, {16.0, 0.90}, {17.0, 2.00}})
```

Isotropic spectra for the ground state

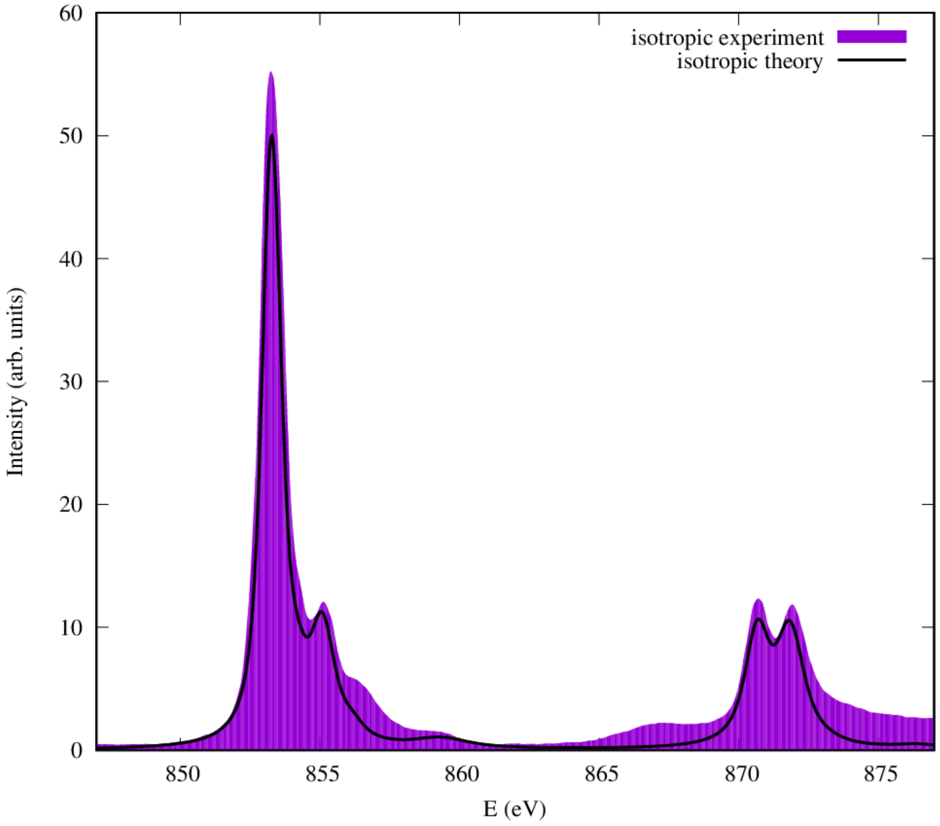
```
XASIsoSpectra = Spectra.Sum(XASSpectra,{1,0,0, 1,0,0,
1,0,0})
```

Write files

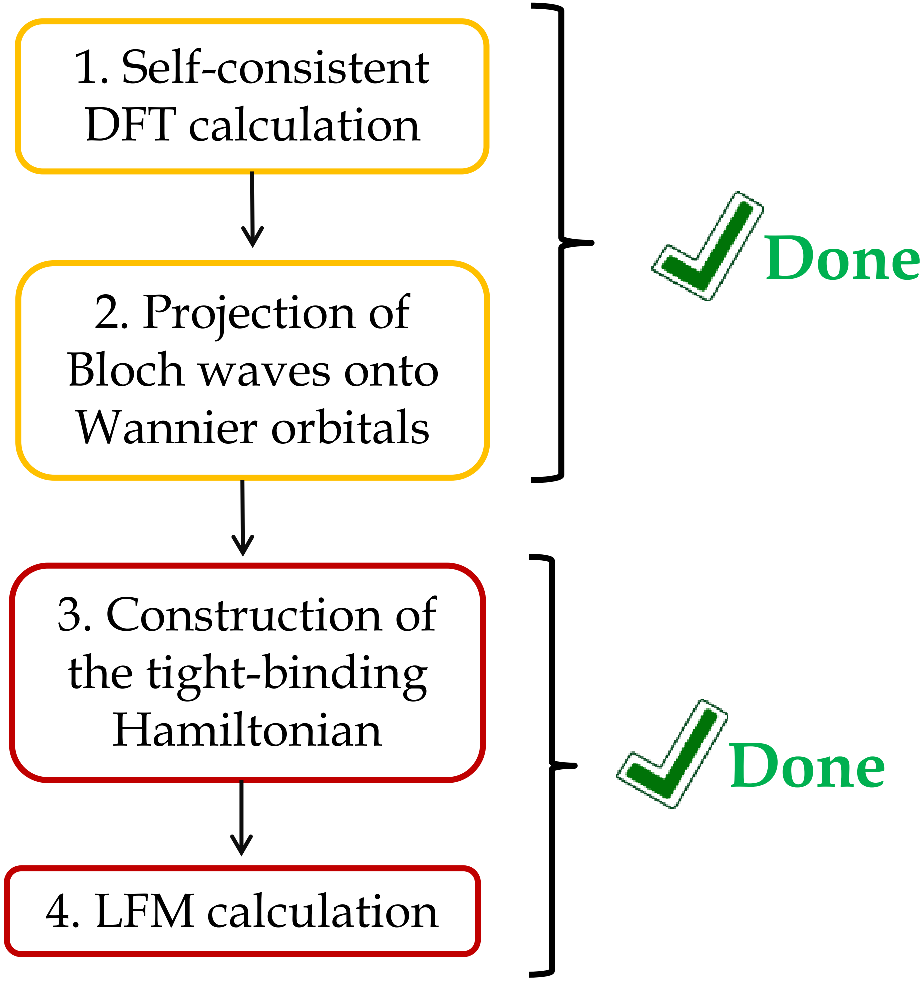
```
XASSpectra.Print({{"file","XASSpec.dat"}})
XASIsoSpectra.Print({{"file","XASIsoSpec.dat"}})
```

Example I: NiO - XAS Calculation

```
$  Quanty 3_XAS.Quanty

$  gnuplot XASSpec.gnuplot
    (generate.ps files)
```



Script method



1. Self-consistent DFT calculation

2. Projection of Bloch waves onto Wannier orbitals

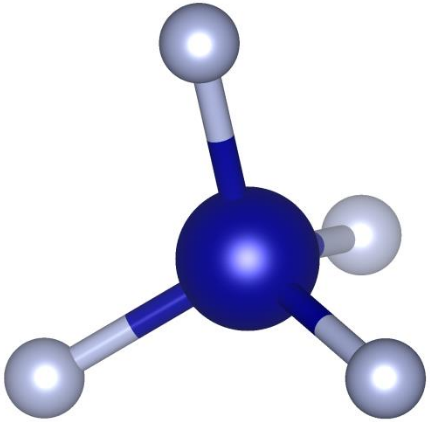
✓ Done

3. Construction of the tight-binding Hamiltonian

4. LFM calculation

✓ Done

Example II: CrF_4 – Ground state calculation



Changes to make from NiO groundstate calculation are listed in **`fromNiOtoCrF4`** in folder **`CrF4/LFMcal`**