Basic post-processing and visualization of results

Stefano Fabris

Theory@Elettra group

CNR-INFM DEMOCRITOS National Simulation Center
Trieste - ITALY

Overview

Basic post-processing and visualization of results

- Molecular and crystal structure visualization
- Electronic band structure
- Density of states
- Charge density

Supercells, forces, and structural optimizations

- Setting up supercells
- Calculation of atomic forces
- Employing forces for obtaining equilibrium geometries
- A working case: supercell slabs for surfaces

Overview

Data analysis and plotting

- Calculating surface energies
- Surface electronic states
- Bonding charge analysis
- Simulating STM images

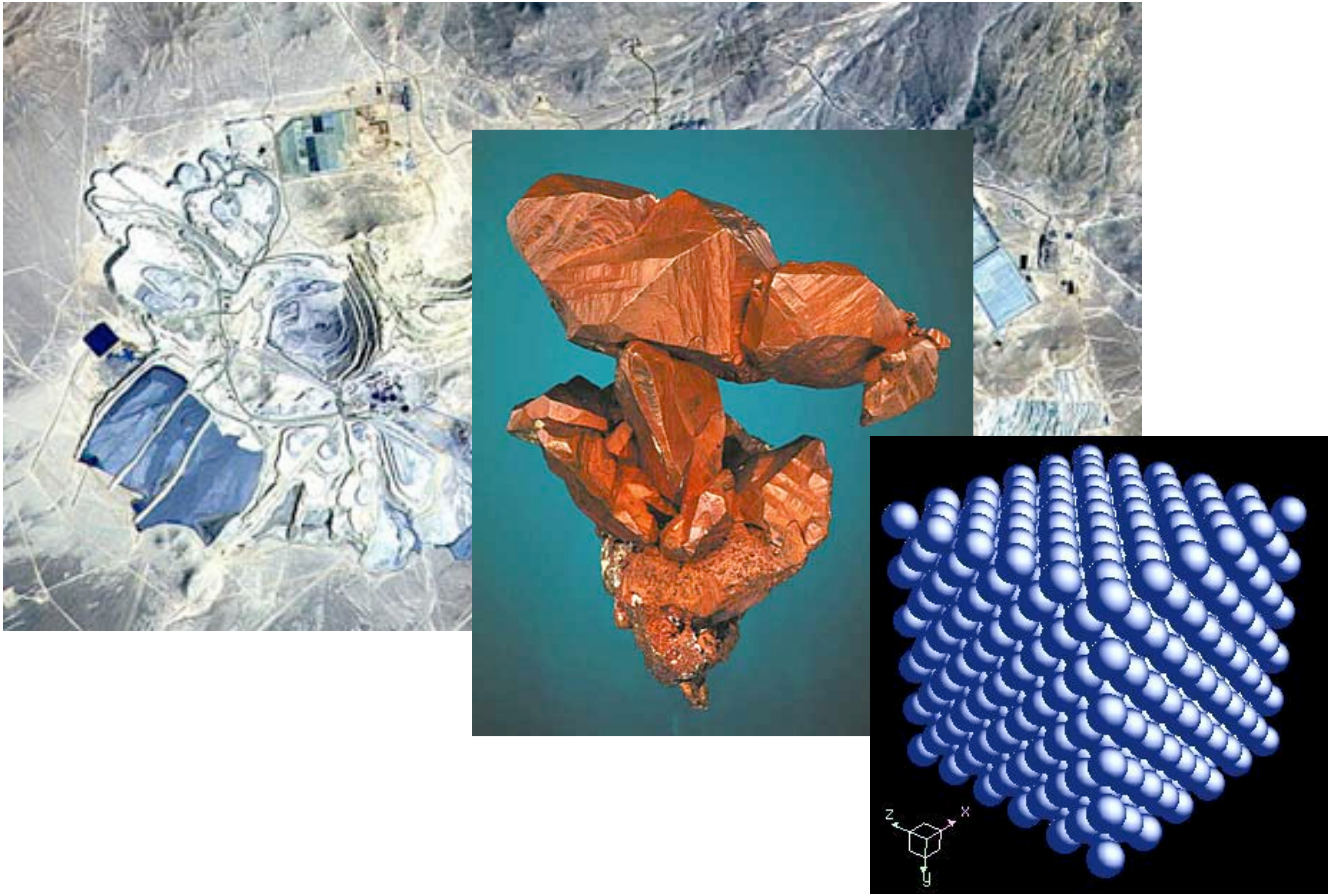
Simulating activated events

- The Nudged Elastic Band method

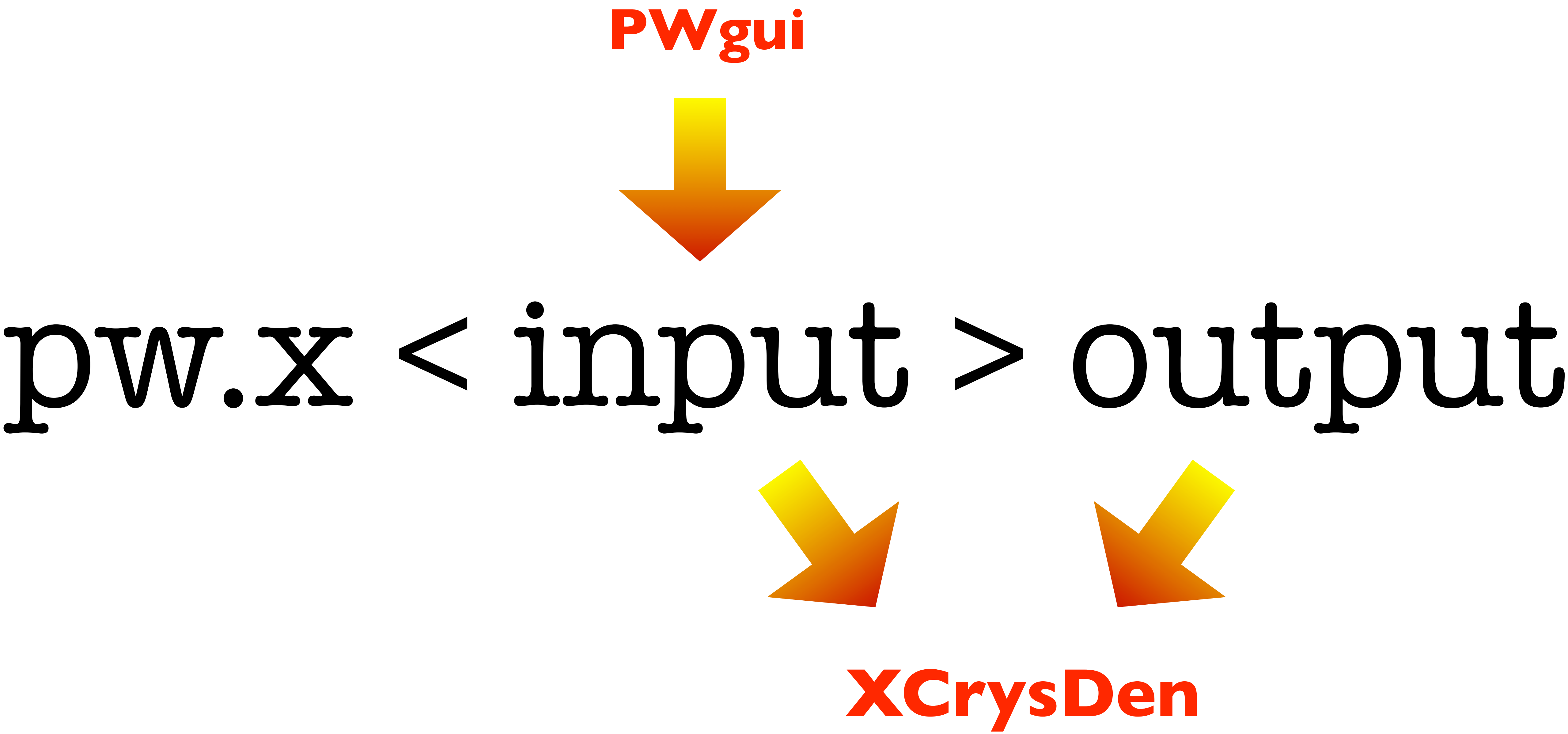
Cu as a working case

Cu as a working case

Cu as a working case



XCrysDen



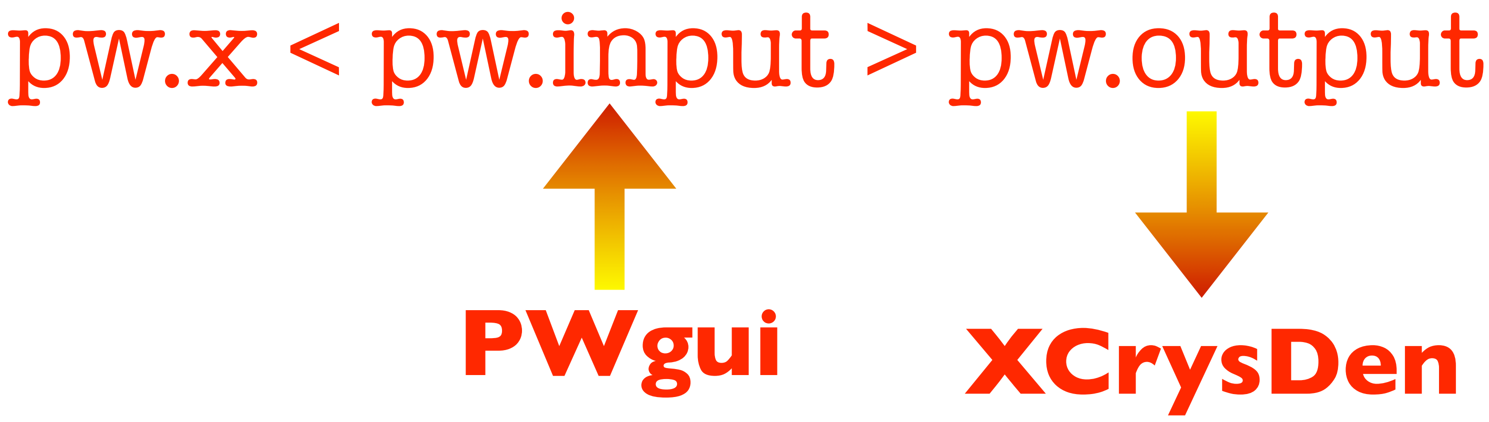
PWgui

```
pw.x < input > output
```

XCrysDen

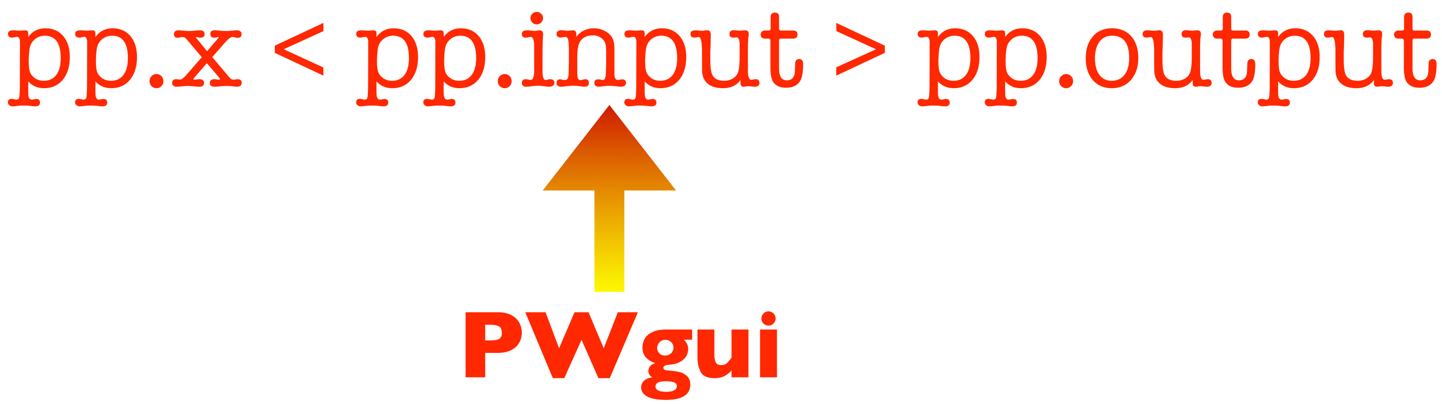
Standard calculation workflow

- ➡ Self-consistent calculation



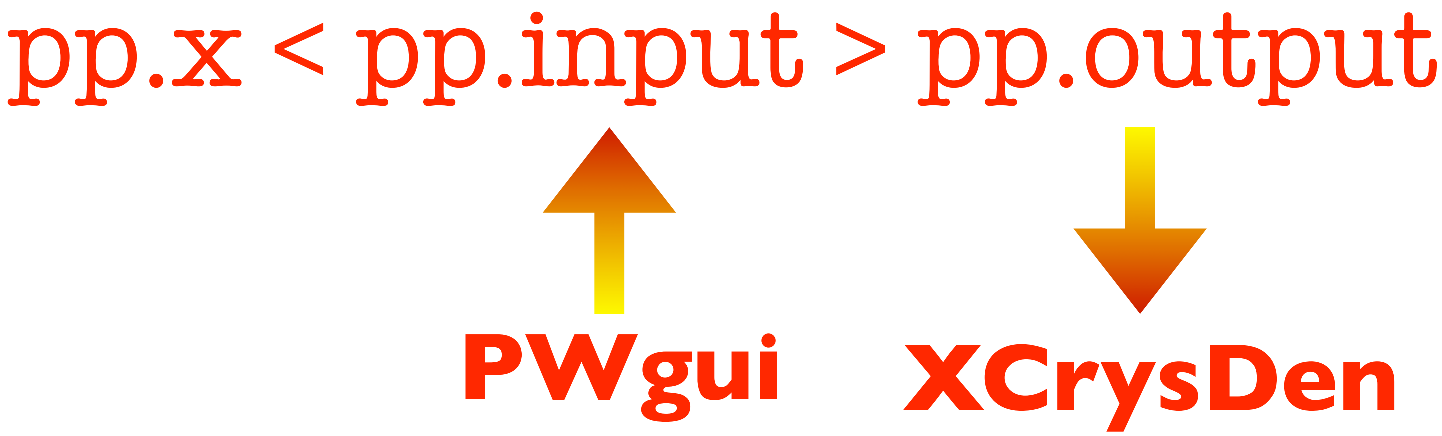
```
pw.x < pw.input > pw.output
```

- ➡ Post processing of data for property calculation (ex. charge density)



```
pp.x < pp.input > pp.output
```

- ➡ Formatting the data for visualization



```
pp.x < pp.input > pp.output
```

XCrysDen

Free software for crystal and molecular structure visualization

XCrySDen ...

X-window CRYstalline Structures and DENsities

Home | About | Description | Documentation | Download | News | Register

- **Home**
  - Mailing list
  - News
  - Terms of use
  - Reference
- **About**
- **Description**
  - Soft. Requirements
- **Documentation**
  - Installation
  - HOWTOs
  - FAQs

XCrySDen

XCrySDen is a **crystalline and molecular structure visualisation program**, which aims at display of isosurfaces and contours, which can be superimposed on crystalline structures and interactively rotated and manipulated. It can run on most *UNIX* platforms, without any special hardware requirements.

XCrySDen has been also ported to MAC OSX (requires X11) and Windows (requires CYGWIN).

The name of the program stands for ***Crystalline Structures and Densities*** and ***X*** because it runs under the X-Window environment.

Read more... | See screenshots ...

XCrySDen mailing list

XCrySDen mailing list is an open mailing list where XCrySDen related issues can be discussed among users.

Subscribe | Archives

www.xcrysden.org

Developed by Anton Kokalj at Josef Stefan Institute - SLO

XCrysDen

Molecular viewer: first steps

➡ Open the file 2.xsf:

`File -> XCrySDen Examples ... -> XSF Files`

➡ Experience on how to

- rotate, zoom, change display mode
- measure distances, angles, dihedral angles
- modify display parameters

XCrysDen

Print crystal structure to file

➡ Select menu “Print”:

`File -> Print`

➡ Set filename and file format (by extension)

- file.png -- PNG format
- file.gif -- GIF format
- fil.eps -- EPS format ... ...

➡ XCrysDen takes a screenshot of your structure. So:

- maximize the XCrysDen window
- enlarge the structure to fill the window
- increase tasselation factor (menu “Modify”) to 40-50

XCrysDen

Display mode: lighting on/off

➡There are two levels of display modes:

- **Lighting-off** mode:
  - very fast but displays only atoms and bonds

- **Lighting-on** mode:
  - fancy and realistic display but may be quite slow

XCrysDen

Crystal viewer: first steps

➡ Open the file ZnS.xsf:

File -> XCrySDen Examples ... -> XSF Files

➡ Experience on how to

- display the crystal cell
- modify the number of displayed cells
- change the unit of repetition
- display Wigner-Seits cell
- display Brillouin Zone

➡ Open the file fcc-410-1x1.xsf:

- display at least 4x4 unit cells
- try:

Tools -> Color scheme -> Slab colors

Ex1: XCrysDen & QE

analyze input/ouput from QE

- ➡ Visualize the crystal structure from input

```
xcrysden --pwi input.file
```

Ex: open the file s110-1x1-7lay.in

- ➡ Visualize the crystal structure from output

```
xcrysden --pwo output.file
```

Ex: open the file pc-cu.rlx.out

Ex1: XCrysDen & QE

Isosurfaces and contours

➡ Open file CO_HOMO.xsf

File -> XCrySDen Examples ... -> XSF Files

➡ Select menu "Data grid"

Tools -> Data Grid

➡ Then:

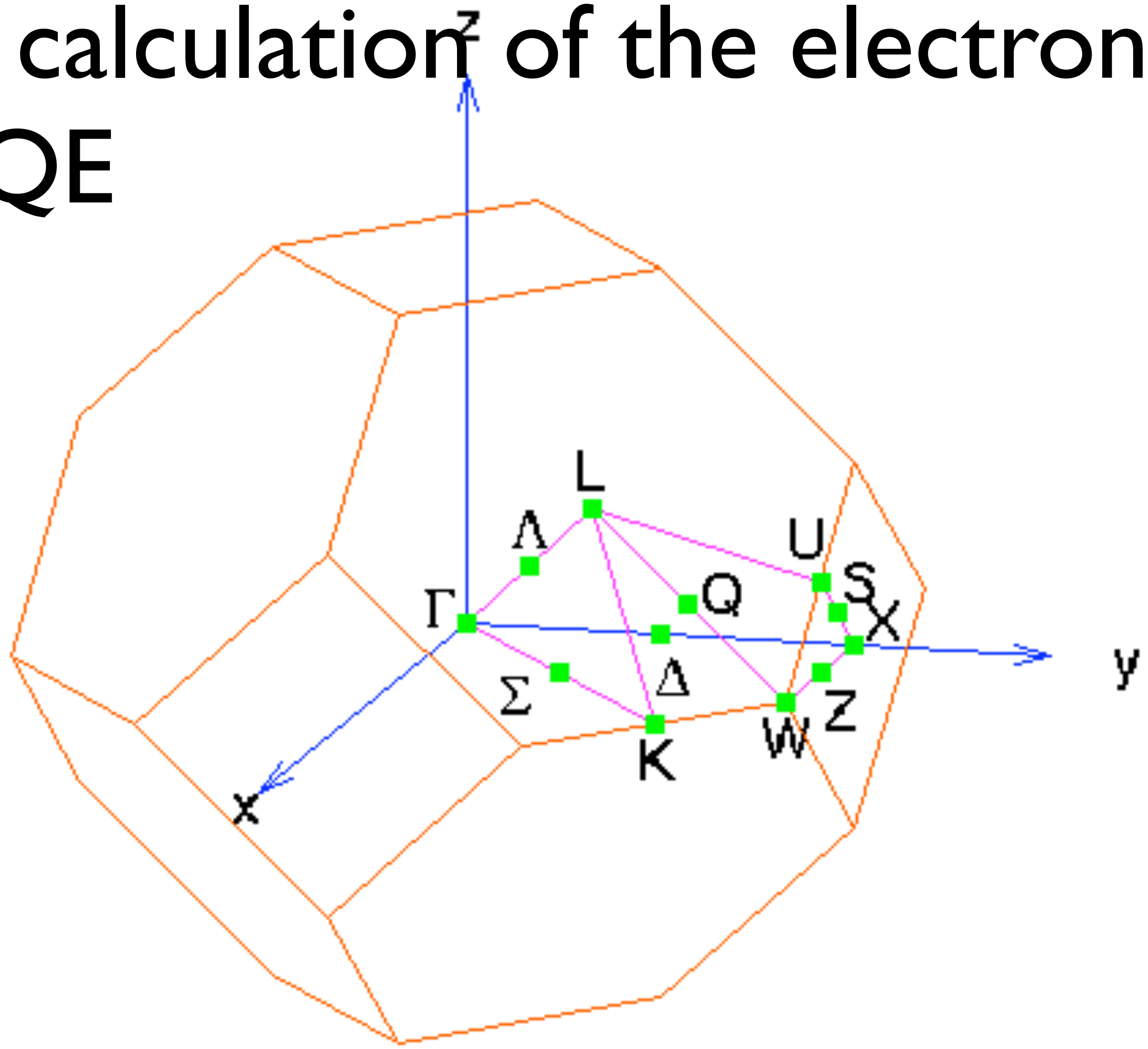
- press [OK]
- specify a value for the isovalue: ex. 0.1 (units??)
- select: "Render +/- isovalue"
- [Submit]

Verify that the HOMO is localized on the C atom and have sigma symmetry

Ex1: XCrysDen & QE

k-path for band-structure calculation

➡ XCrysDen can be used to specify the path in the BZ to be used for the calculation of the electronic band structure with QE



Build the following k-path for a fcc crystal:
Gamma -> X -> U -> L -> Gamma

Ex1: XCrysDen & QE

High-symmetry k-path for a fcc crystal

- Open the file k-path.fcc.in
- Select: Tools -> k-path selection
- select the path by clicking on a sequence of high-symmetry points: Gamma -> X -> U -> L -> Gamma
- specify how many k points to be calculated (Ex: 100)
- save the k-path to file. (.pwscf extension is required for formatting the file for pw.x)

Ex2: Electronic band structure

- ➡ Self-consistent calculation (`pw.x`)

$$\left(-\frac{1}{2}\nabla^2 + V_{\text{eff}}(r)\right)\psi_n(r) = \epsilon_n\psi_n(r)$$

$$\rho(r) = \sum_n |\psi_n(r)|^2 \qquad V_{\text{eff}}(r) = \int \frac{\rho(r')}{|r-r'|}\mathrm{d}r' + V_{XC}(r) + V_{\text{ext}}(r)$$

OUT: SCF density and effective potential

- ➡ Select a k-point path (i.e. by using XCrysDen, ...)

- ➡ Non self-consistent calculation (`pw.x`)

IN: SCF density and effective potential
OUT: set of eigenvalues along the k-point path

- ➡ Plot the eigenvalues along the BZ high-symmetry lines

Band structure of bulk Cu

1) Edit the input file (`run-cu.scf`) and run the scf calculation for bulk Cu (`pw.x`):

```
./run-cu.scf
```

2) Edit the file `run-cu.bands:`

i) choose the "calculation" type for performing a band structure calculation (see &control namelist)

ii) set the number of bands to 10: nbnd=10 (see &system namelist)

iii)specify the k-point list in the input file (use the file .pwscf generated with XCrysDen and insert it after the appropriate card)

3) Run the calculation for the specified set of k-points (`pw.x`):

```
./run-cu.bands
```

Band structure of bulk Cu

4) Prepare an input file for the data post processing program `bands.x` (see `Doc/INPUT_BANDS.txt`). This program print the eigenvalues to the file specified in ”`filband`”

```
&inputpp
 prefix='',
 outdir='',
 filband='YYYY'
/
```

5) Run the postprocessing calculation (`bands.x`):

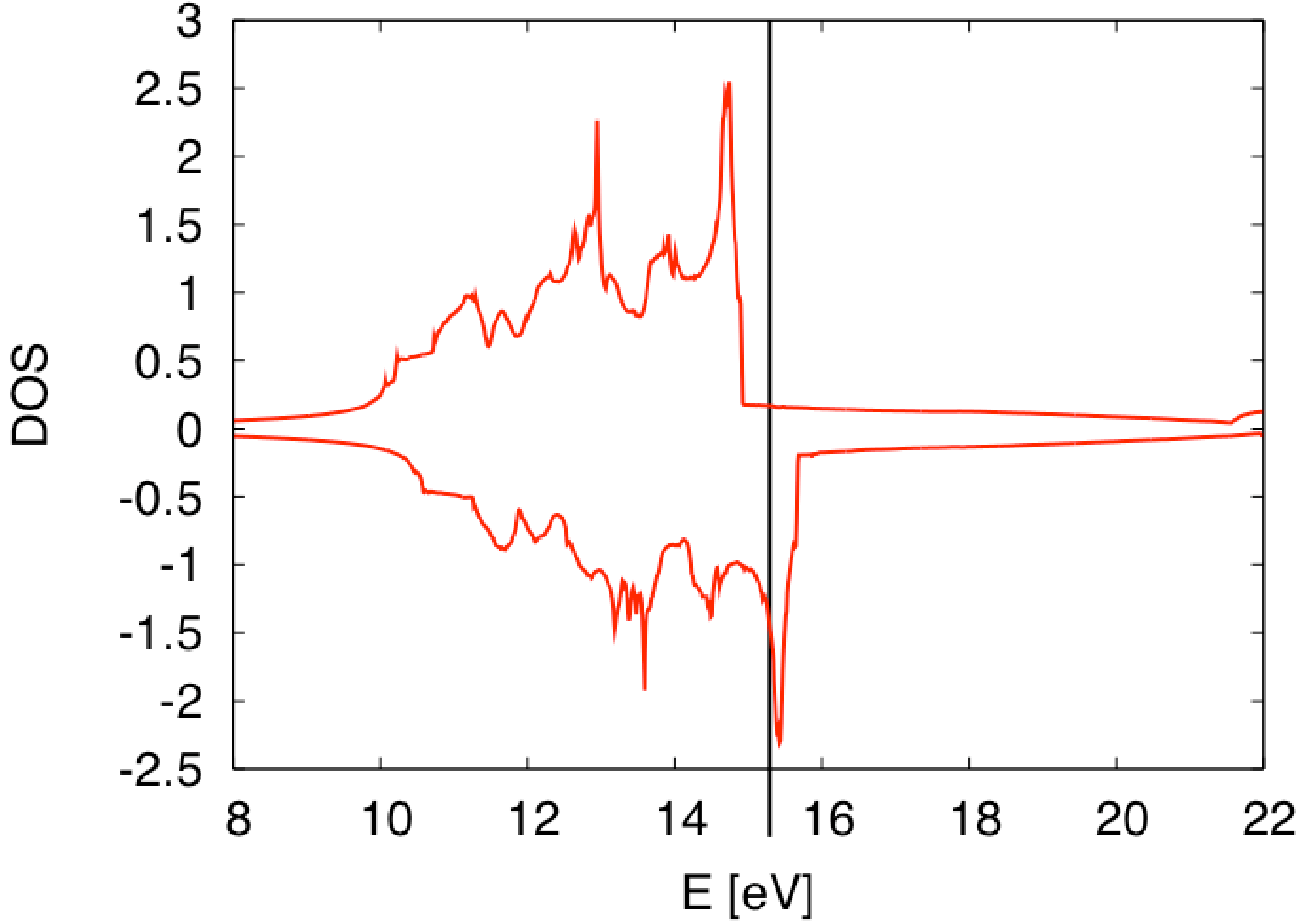
```
bands.x < input > output
```

Band structure of bulk Cu

6) Use the program **`plotband.x`** to obtain a postscript file of the bandstructure and a set of datafiles that can be plotted with your preferred software (gnuplot, xmgr ...).

7) Plot the bandstructure of Cu bulk

Ex3: Density of electronic states - DOS



Ex3: Density of electronic states - DOS

➡ Self-consistent calculation (`pw.x`)

OUT: SCF density and effective potential

➡ Select a finer mesh of k points

➡ Non self-consistent calculation (`pw.x`)

Calculate the KS eigenvalues for the finer mesh of k-points sam by using the SCF effective potential

➡ Post processing (`pp.x`)

Sum all the eigenvalues having the same energy (broadening)

$$D(\varepsilon) = \sum_n \int \delta(\varepsilon - \epsilon_{n\mathbf{k}}) \frac{\mathrm{d}\mathbf{k}}{4\pi^3} \longrightarrow \sum_{n\mathbf{k}} w_{\mathbf{k}}\, \delta(\varepsilon - \epsilon_{n\mathbf{k}})$$

Ex3: Density of electronic states - DOS

1) Edit the input file (**`run-cu.scf`**) and run the scf calculation for bulk Cu (**`pw.x`**):

2) Edit the file **`run-cu.nscf:`**

   i) choose the "calculation" type for performing a non self consistent calculation (see &control namelist)
   ii) set the number of bands to 12: nbnd=12 (see &system namelist)
   iii) set a larger set of K points (Ex: `16 16 16 0 0 0)`

3) Run the nscf calculation (**`pw.x`**):

```
./run-cu.nscf
```

Ex3: Density of electronic states - DOS

4) Prepare an input file for the data post processing program `dos.x` (see `Doc/INPUT_DOS.txt`). This program print the eigenvalues to the file specified in "`fildos`"

```
 &inputpp
     outdir='$TMP_DIR/'
     prefix=''
     fildos=''
     Emin=xxx, Emax=yyy, DeltaE=0.1
 /
```

5) Run the postprocessing calculation (`pp.x`):

```
pp.x < input > output
```

Ex3: Density of electronic states - DOS

6) Inspect the output file generated by the program "**`dos.x`**"

```
#   E (eV)     dos(E)        Int dos(E)
  7.000   0.2438E+00   0.2438E-01
  7.100   0.2389E+00   0.4827E-01
  7.200   0.1856E+00   0.6683E-01
  7.300   0.1152E+00   0.7834E-01
  7.400   0.6089E-01   0.8443E-01
  7.500   0.3635E-01   0.8807E-01
  7.600   0.3682E-01   0.9175E-01
  7.700   0.5050E-01   0.9680E-01
  7.800   0.7013E-01   0.1038E+00
  7.900   0.1017E+00   0.1140E+00
```

Ex3: Density of electronic states - DOS

7) Plot the DOS and the integrated DOS, identify the Fermi level. Compare the position of the d band with respect to the Fermi level. Relate the DOS to the band structure calculated in Ex. 1.

8) Experience the dependency of the resulting DOS on the plotting variables (see Doc/INPUT_DOS.txt):

```
!    ngauss        Type of gaussian broadening (optional)    0
!            =  0  Simple Gaussian (default)
!            =  1  Methfessel-Paxton of order 1
!            = -1  Marzari-Vanderbilt "cold smearing"
!            =-99  Fermi-Dirac function
!    degauss       gaussian broadening, Ry (not eV!)
!    Emin, Emax    min, max energy (eV) for DOS plot
!    DeltaE        energy grid step  (eV)
```

9) Better looking plots of metals can be obtained by using the tetrahedron method for BZ integration in the scf calculation.

Ex. 4: Projected density of electronic states - PDOS

- ➡ Self-consistent calculation `(pw.x)`

OUT: SCF density and effective potential

- ➡ Select a finer mesh of k points

- ➡ Non self-consistent calculation `(pw.x)`

Calculate the KS eigenvalues for the finer mesh of k-points sam by using the SCF effective potential

- ➡ Post processing `(pp.x)`

Sum all the eigenvalues having the same energy with a weight given by the overlap between the corresponding eigenstate and a localized wave function

$$D^{I}_{lm}(\varepsilon) = \sum_{n\mathbf{k}} \left|\langle \psi_{n\mathbf{k}} | \phi^{I}_{lm} \rangle\right|^{2} \; \delta(\varepsilon - \epsilon_{n\mathbf{k}})$$

Ex4: Projected Density of electronic states - PDOS

1) Edit the input file (**`run-ni.scf`**) and run the scf calculation for bulk Ni (**`pw.x`**):

```
./run-ni.scf
```

Does it work? Inspect the output file:

```
more ni.scf.out
```

2) Edit the file **`run-ni.scf`** adding the required information:

```
starting_magnetization(1)=0.0
```

... and run again the simulation. Copy the output file:

```
cp ni.scf.out ni.scf.out-0magn
```

Ex4: Projected Density of electronic states - PDOS

3) Copy the input file:

```
cp run-ni.scf run-ni.nscf
```

4) Edit the file `run-ni.nscf:`

i) choose the "calculation" type for performing a non self consistent calculation (see &control namelist)

ii) set the number of bands to 12: nbnd=12 (see &system namelist)

iii) set a larger set of K points (Ex: `16 16 16 0 0 0)`

5) Run the non self consistent calculation for bulk Ni

```
./run-ni.nscf
```

Ex4: Projected Density of electronic states - PDOS

6) Prepare an input file for the data post processing program `projwfc.x` (see `Doc/INPUT_PROJWFC.txt`). This program print the eigenvalues to the file specified in ”**`filpdos`**”

```
 &inputpp
     outdir='$TMP_DIR'
     prefix='ni'
     filpdos=’xxx’
     Emin=xx, Emax=yy, DeltaE=0.1
     ngauss=1, degauss=0.02
 /
```

7) Run the postprocessing program `projwfc.x`

```
projwfc.x < pdos.input > pdos.output
```

Ex4: Projected Density of electronic states - PDOS

8) Inspect the output files generated by `projwfc.x`

```
 more pdos.out
...  ...  ...
Lowdin Charges:

     Atom #   1: total charge =   9.8845,
s =  0.9326, p =  0.0000, d =  8.9519,

     Spilling Parameter:   0.0116
```

Don’t you notice anything unusual or surprising?
What is the occupation of the Ni s and d states that you would have expected?

Ex4: Projected Density of electronic states - PDOS

9) Edit the file `run-ni.scf` and modify the starting magnetization:

`starting_magnetization(1)=0.7`

10) Run again the scf calculation (`pw.x`).

11) Compare the total energy of this simulation with the one obtained before (see file `ni.scf.out-0magn`). Do they differ? Why (or why not)?

12) Check the spin polarization of the two simulations.

13) Recalculate the pdos for the latter SCF solution (follow again steps 3-7, set a different name for "`filpdos`")

Ex4: Projected Density of electronic states - PDOS

14) Inspect the output files generated by `projwfc.x`

```
more pdos.out
...  ...  ...
Lowdin Charges:
     Atom #   1: total charge =    9.8825, s =  0.9368, p =  0.0000,
d =  8.9457,

                 spin up      =    5.2311, s =  0.4609, p =  0.0000,
d =  4.7702,

                 spin down    =    4.6515, s =  0.4759, p =  0.0000,
d =  4.1756,

                 polarization =    0.5796, s = -0.0150, p =  0.0000,
d =  0.5946,
```

Does this result make more sense? Why?

```
more ni.pdos_atm#1(Ni)_wfc#1(s)
```

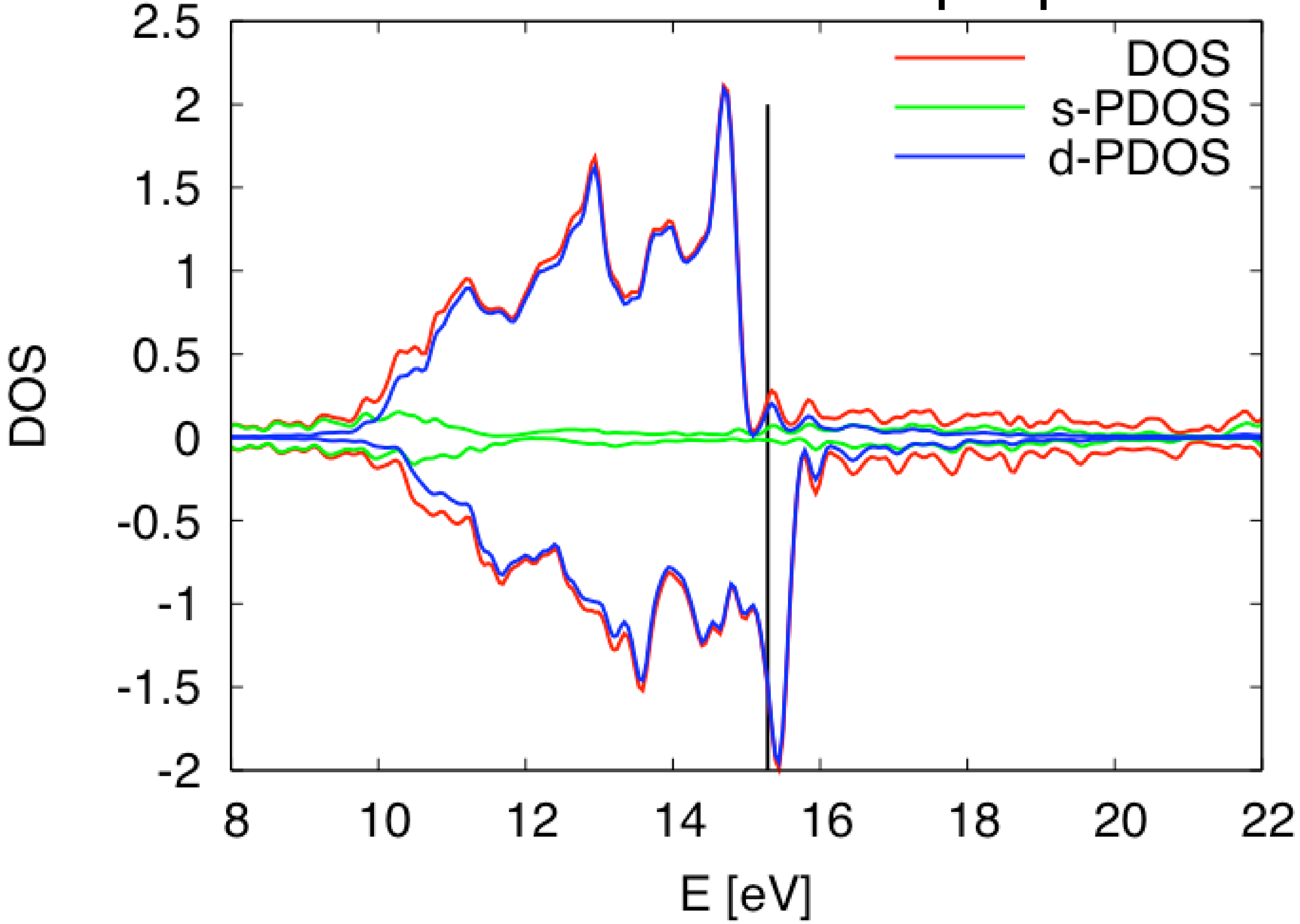
```
... ... ...
# E (eV)   ldosup(E)   ldosdw(E) pdosup(E)   pdosdw(E)
  5.000 -0.263E-05 -0.152E-05 -0.263E-05 -0.152E-05
  5.100 -0.133E-04 -0.840E-05 -0.133E-04 -0.840E-05
  5.200 -0.513E-04 -0.351E-04 -0.513E-04 -0.351E-04
  5.300 -0.152E-03 -0.112E-03 -0.152E-03 -0.112E-03
  5.400 -0.338E-03 -0.274E-03 -0.338E-03 -0.274E-03
  5.500 -0.492E-03 -0.470E-03 -0.492E-03 -0.470E-03
```

```
more ni.pdos_atm#1(Ni)_wfc#2(d)
```

```
# E (eV)   ldosup(E)   ldosdw(E) pdosup(E)   pdosdw(E)   pdosup(E)
pdosdw(E)   pdosup(E)   pdosdw(E)   pdosup(E)   pdosdw(E)   pdosup(E)
pdosdw(E)
  5.000 -0.704E-10 -0.294E-10 -0.510E-11 -0.209E-11 -0.201E-10
-0.840E-11 -0.201E-10 -0.840E-11 -0.510E-11 -0.209E-11 -0.201E-10
-0.840E-11
  5.100 -0.664E-09 -0.303E-09 -0.575E-10 -0.258E-10 -0.183E-09
-0.839E-10 -0.183
E-09 -0.839E-10 -0.575E-10 -0.258E-10 -0.183E-09 -0.839E-10
... ... ...
```

Ex4: Projected Density of electronic states - PDOS

15) Plot the DOS and PDOS of bulk Ni: spin-polarized ground state



16) Compare it with the PDOS of the spin-unpolarized metastable state

Ex5: Charge density analysis

1) Edit the input file (**`run-si.scf`**) and run the scf calculation for bulk Si (**`pw.x`**):

```
./run-si.scf
```

2) The charge density is calculated by means of the post processing program **`pp.x`** . This program can perform several operations: check the documentation **`Doc/INPUT_PROJWFC.txt`**

2)Prepare an input file for the data post processing program **`pp.x`** .

```
  !      Program: pp.x
  !      Purpose: data analysis and plotting.
  !      The code performs two steps:
  !      1) reads the output file produced by pw.x, extract and
calculate
  !         the desired quantity (rho, V, ...)
  !      2) writes the desired quantity to file in a suitable format
for
  !         various types of plotting and various plotting programs
  !
  !      The input data of this program are read from standard input
  !      or from a file and have the following format:
  !
  ! Namelist &inputpp
  !      containing the variables for step 1), followed by
  ! Namelist &plot
  !      containing the variables for step 2)
!-&inputPP          Namelist &inputPP; contains
  !
  ! prefix      prefix of files saved by program pw.x
  ! outdir      temporary directory where pw.x files resides
  !
  ! filplot     file "filplot" contains the quantity selected by
plot_num
  !             (can be saved for further processing)
```

```
! plot_num    selects what is saved in filplot:
  !                   0=charge
  !                   1=total potential V_bare+V_H + V_xc
  !                   2=local ionic potential
  !                   3=local density of states at e_fermi
  !                   4=local density of electronic entropy
  !                   5=STM images
  !                   6=spin polarization (rho(up)-rho(down))
  !                   7=|psi|^2
  !                   8=electron localization function (ELF)
  !                   9=planar average of all |psi|^2
  !                  10=integrated local density of states (ILDOS)
  !                     from emin to emax (emin, emax in eV)
  !                     if emax is not specified, emax=E_fermi
  !                  11=the V_bare + V_H potential
  !                  12=the electric field potential
  !                  13=the noncollinear magnetization.
  !
  !               Options for total charge (plot_num=0):
  !
  ! spin_component 0=total charge (default value),
  !                   1=spin up charge,
  !                   2=spin down charge.
```

Ex5: Charge density analysis

3) Prepare an input file for the data post processing program **`pp.x`** .

```
&inputpp
    prefix  = 'si'
    outdir  = '$TMP_DIR',
    filplot = 'si.charge'
    plot_num= XX
 /
```

4) Run the postprocessing program **`pp.x`**

```
pp.x < charge.input > charge.output
```

The output file `'si.charge'` contains the values of the total valence charge on a regular grid in real space

The output file `'si.charge'` contains the values of the total valence charge on a regular grid in real space. This data file can now be further processed for plotting some specific property.

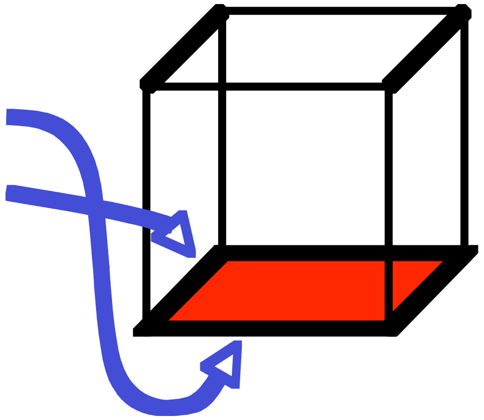
```
  !-&plot            Namelist &plot; contains
  !
  !      nfile        the number of data files (OPTIONAL, default: 1)
  !
  !----FOR i = 1, nfile:
  !
  !      filepp(i)    nfile=1: file containing the quantity to be plotted
  !                   nfile>1: see "weight"
  !                   (default: filepp(1)=filplot)
  !      weight(i)    weighing factors: assuming that rho(i) is the
quantity
  !                   read from filepp(i), the quantity that will be
plotted is:
  !                   weight(1)*rho(1) + weight(2)*rho(2) +
weight(3)*rho(3)+...
  !                   (OPTIONAL: default weight(1)=1.0)
  !
  !                   BEWARE: atomic coordinates are read from the first
file;
  !                   if their number is different for different files,
  !                   the first file must have the largest number of atoms
  !
  !----END_FOR
```

```
  !      iflag     0 1D plot of the spherical average
  !                1 1D plot
  !                2 2D plot
  !                3 3D plot
  !                4 2D polar plot on a sphere
  !
  !      output_format  (ignored on 1D plot)
  !                0  format suitable for gnuplot   (1D)
  !                1  format suitable for contour.x (2D)
  !                2  format suitable for plotrho   (2D)
  !                3  format suitable for XCRYSDEN  (1D, 2D, 3D)
  !                4  format suitable for gOpenMol  (3D)
  !                   (formatted: convert to unformatted *.plt)
  !                5  format suitable for XCRYSDEN  (3D)
  !                6  format as gaussian cube file  (3D)
  !                   (can be read by many programs)
...
  !-ELSEIF iflag = 2
  !                   REQUIRED:
  !          e1, e2  3D vectors which determine the plotting plane
  !                   (must be orthogonal)
  !              x0  3D vector, origin of the plane
  !          nx, ny  number of points in the plane:
  !                  rho(i,j) = rho( x0 + e1 * (i-1)/(nx-1)
  !                                    + e2 * (j-1)/(ny-1) ), i=1,nx ;
j=1,ny
  !
  !      fileout   name of the file to which the plot is written
  !                (DEFAULT: standard output)
```

Ex5: Charge density analysis

5) Prepare an input file (for the program `pp.x`) that reads the file `'si.charge'` and that plot a 2D charge density on a (001) plane intersecting the Si atoms. Specify the output format for XCrysDen.

```
&plot
  nfile=x
  filepp(1)='si.charge'
  iflag=x
  output_format=x
  e1(1)=xx, e1(2)=xx, e1(3)=xx,
  e2(1)=xx, e2(2)=xx, e2(3)=xx,
  x0(1)=xx, x0(2)=xx, x0(3)=xx,
  nx=40, ny=40
  fileout='si.charge001.xsf'
 /
```



001 plane

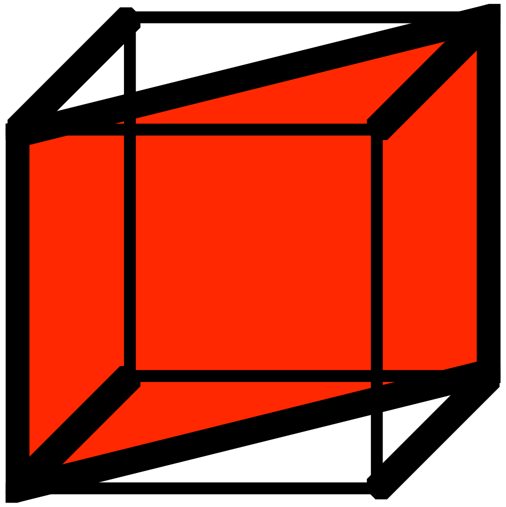
Ex5: Charge density analysis

6) Plot the charge density with XCrysDen and save it to file

Ex5: Charge density analysis

5) Prepare an input file (for the program `pp.x`) that reads the file `'si.charge'` and that plot a 2D charge density on a (110) plane intersecting the Si atoms. Specify the output format for XCrysDen.

```
&plot
   nfile=x
   filepp(1)='si.charge'
   iflag=x
   output_format=x
   e1(1)=xx, e1(2)=xx, e1(3)=xx,
   e2(1)=xx, e2(2)=xx, e2(3)=xx,
   x0(1)=xx, x0(2)=xx, x0(3)=xx,
   nx=40, ny=40
   fileout='si.charge001.xsf'
 /
```



Ex5: Charge density analysis

6) Plot the charge density with XCrysDen and save it to file as a png image