

XSPECTRA

A tool for X-ray absorption spectra (XAS) calculations

G. Radtke* & N. Mas*,[‡]

**Institut de Minéralogie, de Physique des Matériaux et de Cosmochimie,
Université Pierre et Marie Curie, Sorbonne Universités*

[‡]Synchrotron SOLEIL

Slides courtesy of O. Bunău, M. Calandra and D. Cabaret



EWinS 2016 - EUSpec Winter School on Core level spectroscopies
Ajdovščina, Slovenia,
February 5th 2016

1. Introduction on XAS

2. The PAW theory

3. Hands on (I) : Si-K edge in α -quartz SiO_2

- GIPAW pseudopotential generation for Si and Si1s*
- SCF calculation for α -quartz SiO_2 unit cell without/with core-hole
- core wave-function extraction

4. The Lanczos algorithm

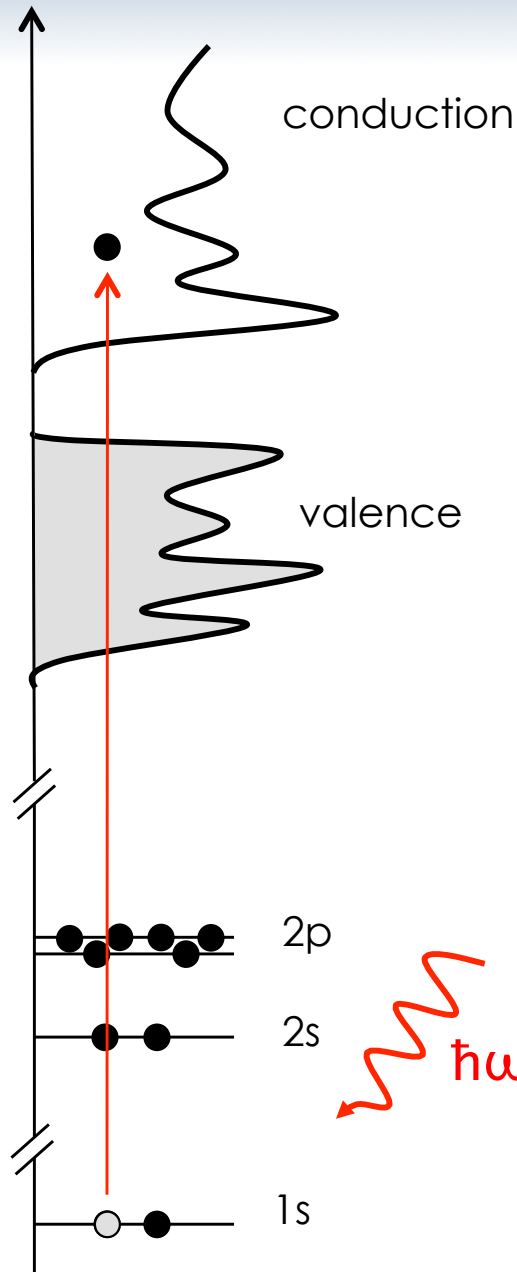
5. Hands on (II) : Si-K edge in α -quartz SiO_2

- XSpectra calculations for α -quartz SiO_2 unit cell without/with core-hole
- Supercell calculation : 2x2x2 cell for α -quartz SiO_2

6. Hands on (III) : Dipolar and Quadrupolar contributions at the Ni-K edge in NiO

7. Summary

A probe of the projected density of unoccupied electronic states



The absorption cross-section reads :

$$\sigma(\omega) = 4\pi^2 \alpha_0 \hbar \omega \sum_{i,f} |\langle \psi_f | \mathcal{O} | \psi_i \rangle|^2 \delta(E_f - E_i - \hbar\omega)$$

Prefactor

Final one-electron state (atomic state, molecular orbital, Bloch state, ...)

Transition operator (electric dipole, ...)

Initial one-electron (core-state)

Energy conservation of the system (specimen + photon)

Assuming a transition operator of the form (electric dipole approximation) :

$$\mathcal{O} \approx \hat{\epsilon} \cdot \mathbf{r}$$

The angular part of the matrix element leads to :

$$l_f = l_i \pm 1$$

In a mono-electronic picture (weak $e^- - e^-$ interaction) :

K (or L_1) edges probe p -LDOS, L_{23} edges probe $s+d$ -LDOS, M_{45} edges probe $p+f$ -LDOS, ...

What is it ?

XSpectra calculates X-ray absorption dipolar and quadrupolar cross sections in the pre-edge to near-edge region within the single particle approximation (i.e. mostly K or L₁ edges but also certain L₂₃ edges)

Where can I find XSpectra ?

XSpectra is distributed in the QUANTUM-ESPRESSO package (<http://www.quantum-espresso.org/>)

Can I use it ?

XSpectra is distributed under the GNU licence, so you can use it for free. Please remember to cite the following papers to acknowledge people building the software:

- P. Giannozzi *et al.*, *J. Phys. Condens. Matter* **21**, 395502 (2009).
- C. Gougoussis, M. Calandra, A. P. Seitsonen and F. Mauri, *Phys. Rev. B* **80**, 075102 (2009)
- M. Taillefumier, D. Cabaret, A. M. Flank and F. Mauri, *Phys. Rev. B* **66**, 195107 (2002)

Outline

1. Introduction on XAS
- 2. The PAW theory**
3. Hands on (I) : Si-K edge in α -quartz SiO_2
 - GIPAW pseudopotential generation for Si and $\text{Si}1s^*$
 - SCF calculation for α -quartz SiO_2 unit cell without/with core-hole
 - core wave-function extraction
4. The Lanczos algorithm
5. Hands on (II) : Si-K edge in α -quartz SiO_2
 - XSpectra calculations for α -quartz SiO_2 unit cell without/with core-hole
 - Supercell calculation : $2 \times 2 \times 2$ cell for α -quartz SiO_2
6. Hands on (III) : Dipolar and Quadrupolar contributions at the Ni-K edge in NiO
7. Summary

The PAW (Projected Augmented Wave) method

We need to reconstruct the *all electron* (AE) states from *pseudo* (PS) states
(Note : AE states do not mean many-electron states !!!)

The problem arises for $|\psi_f\rangle$ in the absorption cross-section, as we only get $|\tilde{\psi}_f\rangle$ from the PP-PW calculation. $|\psi_i\rangle$ is an atomic core-state.

We admit that a linear transformation \mathcal{T} exists, such that :

$$|\tilde{\psi}\rangle \xleftrightarrow{\text{mapping}} |\psi\rangle = \mathcal{T}|\tilde{\psi}\rangle$$

P. E. Blöchl, *Phys. Rev. B* **50**, 17953 (1994)

The PAW (Projected Augmented Wave) method

\mathcal{T} is linear

\mathcal{T} differs from identity in the core (augmentation) regions $\Omega_{\mathbf{R}}$ only :

$$\mathcal{T} = \mathbf{1} + \sum_{\mathbf{R}} \mathcal{T}_{\mathbf{R}} = \mathbf{1} + \sum_{\mathbf{R},n} \left(|\phi_{\mathbf{R},n}\rangle - |\tilde{\phi}_{\mathbf{R},n}\rangle \right) \langle \tilde{p}_{\mathbf{R},n}|$$

sum over atomic sites,
(ℓ, m) and eventually,
a channel index j

all electron partial waves

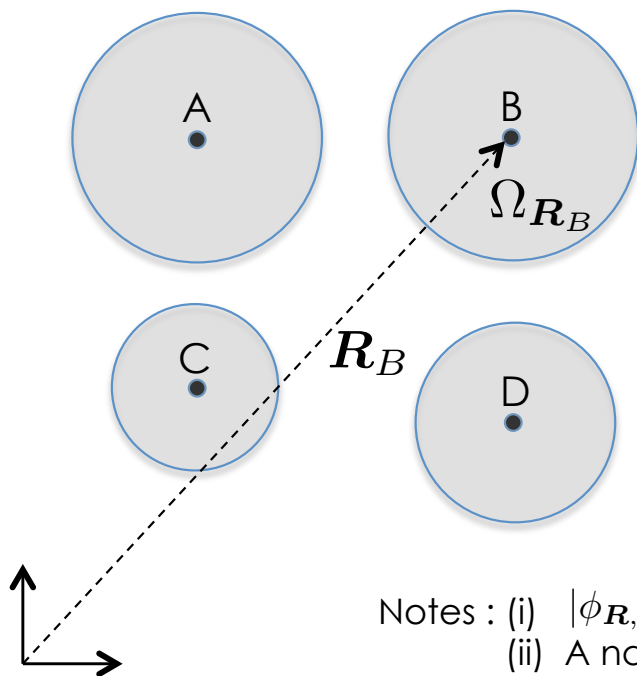
pseudo partial waves

Projection functions (one per (ℓ, m) channel)

(i) $|\tilde{p}_{\mathbf{R},n}\rangle$ are 0 outside $\Omega_{\mathbf{R}}$

(ii) Inside $\Omega_{\mathbf{R}}$, $\langle \tilde{p}_{\mathbf{R},n} | \tilde{\phi}_{\mathbf{R}',n'} \rangle = \delta_{\mathbf{R}\mathbf{R}'} \delta_{nn'}$

$$\text{or } \sum_n |\tilde{p}_{\mathbf{R},n}\rangle \langle \tilde{\phi}_{\mathbf{R},n}| = \mathbf{1}$$



Notes : (i) $|\phi_{\mathbf{R},n}\rangle$ and $|\tilde{\phi}_{\mathbf{R},n}\rangle$ coincide outside $\Omega_{\mathbf{R}}$
 (ii) A natural choice for $|\phi_{\mathbf{R},n}\rangle$ are the solution of the radial Schrödinger equation for the isolated atom

Cross-section in the PAW formalism

Using $|\psi_f\rangle = \mathcal{T}|\tilde{\psi}_f\rangle$, we get :

$$\langle\psi_f|\mathcal{O}|\psi_i\rangle = \langle\tilde{\psi}_f|\mathcal{O}|\psi_i\rangle + \sum_{\mathbf{R},n} \langle\tilde{\psi}_f|\tilde{p}_{\mathbf{R},n}\rangle \langle\phi_{\mathbf{R},n}|\mathcal{O}|\psi_i\rangle - \sum_{\mathbf{R},n} \langle\tilde{\psi}_f|\tilde{p}_{\mathbf{R},n}\rangle \langle\tilde{\phi}_{\mathbf{R},n}|\mathcal{O}|\psi_i\rangle$$

Here, $|\psi_i\rangle$ is localized on the site of the absorbing atom \mathbf{R}_0 so that all matrix elements will be negligible except for $\mathbf{R} = \mathbf{R}_0$

$$\langle\psi_f|\mathcal{O}|\psi_i\rangle \approx \langle\tilde{\psi}_f|\mathcal{O}|\psi_i\rangle + \sum_n \langle\tilde{\psi}_f|\tilde{p}_{\mathbf{R}_0,n}\rangle \langle\phi_{\mathbf{R}_0,n}|\mathcal{O}|\psi_i\rangle - \sum_n \langle\tilde{\psi}_f|\tilde{p}_{\mathbf{R}_0,n}\rangle \langle\tilde{\phi}_{\mathbf{R}_0,n}|\mathcal{O}|\psi_i\rangle$$

Using $\sum_n |\tilde{p}_{\mathbf{R},n}\rangle \langle\tilde{\phi}_{\mathbf{R},n}| = \mathbf{1}$, the first and last terms cancel, so that :

$$\langle\psi_f|\mathcal{O}|\psi_i\rangle \approx \sum_n \langle\tilde{\psi}_f|\tilde{p}_{\mathbf{R}_0,n}\rangle \langle\phi_{\mathbf{R}_0,n}|\mathcal{O}|\psi_i\rangle$$

Defining $|\tilde{\varphi}_{\mathbf{R}_0}\rangle = \sum_n |\tilde{p}_{\mathbf{R}_0,n}\rangle \langle\phi_{\mathbf{R}_0,n}|\mathcal{O}|\psi_i\rangle$, the cross-section finally reads :

$$\sigma(\omega) = 4\pi^2 \alpha_0 \hbar \omega \sum_f |\langle\tilde{\psi}_f|\tilde{\varphi}_{\mathbf{R}_0}\rangle|^2 \delta(E_f - E_i - \hbar\omega)$$

M. Taillefumier et al. , *Phys. Rev. B* **66**, 195107 (2002)

Practical PAW for XAS

In the expression :

$$|\tilde{\varphi}_{\mathbf{R}_0}\rangle = \sum_n |\tilde{p}_{\mathbf{R}_0,n}\rangle \langle \phi_{\mathbf{R}_0,n} | \mathcal{O} | \psi_i \rangle$$

The sum runs over a complete set, *i.e.* an infinite number of projectors !!

In practice a **finite** number of projectors is enough :

- 1 projector/channel (ℓ)
 - ➡ generally yields **wrong intensities**
 - ➡ **wrong dipole/quadrupole ratio**
- 2 projectors/channel (ℓ)
 - ➡ **correct intensities** in the near edge region (~ 50 eV above the edge, in most of the cases)
 - ➡ need to be linearly independent (*i.e.* , span a 2 x 2 subspace)

Finally, we need the core WF **without** the core-hole and the AE partial-waves (from PP generation) to fully determine $|\tilde{\varphi}_{\mathbf{R}_0}\rangle$

Let's go for an example : the Si and the Si1s* ultrasoft PPs

Outline

1. Introduction on XAS
2. The PAW theory
3. Hands on (I) : Si-K edge in α -quartz SiO_2
 - GIPAW pseudopotential generation for Si and Si1s*
 - SCF calculation for α -quartz SiO_2 unit cell without/with core-hole
 - core wave-function extraction
4. The Lanczos algorithm
5. Hands on (II) : Si-K edge in α -quartz SiO_2
 - XSpectra calculations for α -quartz SiO_2 unit cell without/with core-hole
 - Supercell calculation : 2x2x2 cell for α -quartz SiO_2
6. Hands on (III) : Dipolar and Quadrupolar contributions at the Ni-K edge in NiO
7. Summary

Practical details

```
cp -r ~/TutorialXSpectra /scratch/  
cd /scratch/TutorialXSpectra/
```

Directory structure:

./pseudo/	pseudopotentials for this tutorial
./PPgeneration/	input files necessary to generate a GIPAW pseudopotential for Si
./SiO2	input files to be modified for the examples
./SiO2h	
./SiOsupercell	
./NiO	
./*/outdir/	tmp output
./solutions/	reference inputs and outputs
./references/	relevant papers, the .pdf of these lecture and manual INPUT_XSPECTRA

Calculation flow for XAS with XSpectra

Prepare the GIPAW pseudopotentials



Extract the core wavefunction



Prepare input file and run SCF calculation



Prepare input file and run XSpectra

GIPAW (Gauge independent PAW pseudopotentials)

The GIPAW pseudopotential includes all the reconstruction information needed to run XSPECTRA

- needed for the absorbing atom only (non-absorbing atoms accept any kind of pseudopotential)
- contains the following information on the absorbing atom:
 - the core wavefunction without hole
 - the *all electron* partial waves atomic states
 - the Blöchl projectors
- can be obtained with the atomic code `ld1.x`

Generation of (GI)PAW pseudopotential

We want to study the Si K-edge in α -quartz (SiO_2) in the electric dipole approximation. This corresponds to electronic transitions :

$1s$ core-state  unoccupied p states

We only need a PP with two p projectors. We add the s projectors as an exercise even if they are not required

	Projector channel	First projector Energy	Second projector Energy
optional	s	3s state	4s state
mandatory	p	3p state	4p state

Remember that a minimum of 2 projectors/channel is needed !!

Generation of (GI)PAW pseudopotential

```
$ cd PPgeneration  
$ gedit Si.ld1.in
```

```
&input  
  title='Si',  
  zed=14.,  
  rel=1,  
  config='1s2 2s2 2p6 3s2 3p2 3d-2 4s0 4p0',  
  iswitch=3,  
  dft='PBE`  
/  
...
```

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
&input  
  title='Si',  
  zed=14.,  
  rel=1,  
  config='1s2 2s2 2p6 3s2 3p2 3d-2 4s0 4p0',  
  iswitch=3,  
  dft='PBE`  
/  
...
```

Atomic number

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
&input  
  title='Si',  
  zed=14.,  
  rel=1,  
  config='1s2 2s2 2p6 3s2 3p2 3d-2 4s0 4p0',  
  iswitch=3,  
  dft='PBE`  
/  
...
```

Electronic configuration of the isolated atom.



3s, 3p, 4s and 4p states **must be included** because we want to generate projectors at their energies.

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
&input
  title='Si',
  zed=14.,
  rel=1,
  config='1s2 2s2 2p6 3s2 3p2 3d-2 4s0 4p0',
  iswitch=3,
  dft='PBE'
/
...
```

Exchange-correlation functionnal (PZ, PW91, BP, PBE ...)



The pseudopotentials for all atoms in the calculation must have the **same functionnal**.

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
&inputp  
  pseudotype=3,  
  file_pseudopw='Si.pbe-us_gipaw.UPF',  
  lloc=2,  
  tm=.true.  
  which_augfun='PSQ'  
  rmatch_augfun=1.8  
  nlcc=.true.,  
  new_core_ps=.true.,  
  rcore=1.3,  
  lgipaw_reconstruction=.true.
```

/

...

Type of the pseudopotential:

- 2 for Norm-conserving
- 3 for Ultrasoft

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
&inputp
  pseudotype=3,
  file_pseudopw='Si.pbe-us_gipaw.UPF',
  lloc=2,
  tm=.true.
  which_augfun='PSQ'
  rmatch_augfun=1.8
  nlcc=.true.,
  new_core_ps=.true.,
  rcore=1.3,
  lgipaw_reconstruction=.true.
```

/

...

File where the generated PP is written

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
&inputp
  pseudotype=3,
  file_pseudopw='Si.pbe-us_gipaw.UPF',
  lloc=2,
  tm=.true.
  which_augfun='PSQ'
  rmatch_augfun=1.8
  nlcc=.true.,
  new_core_ps=.true.,
  rcore=1.3,
  lgipaw_reconstruction=.true.
```

/

...

Flag to generate pseudo-potentials containing **GIPAW information**

Generation of (GI)PAW pseudopotential

File `Si.ld1.in`

```
5
3S 1 0 2.00 0.00 2.00 2.10 0.0
3S 1 0 0.00 6.00 1.40 2.10 0.0
3P 2 1 2.00 0.00 2.00 2.10 0.0
3P 2 1 0.00 6.00 1.40 2.10 0.0
3D 3 2 -2.00 -0.30 2.00 2.00 0.0
```

Number of wave functions
to be pseudized

```
/
&test
/
```

```
4
3S 1 0 2.00 0.00 2.00 2.10 0.0
4S 2 0 0.00 0.00 2.00 2.10 0.0
3P 2 1 2.00 0.00 1.40 2.10 0.0
4P 3 1 0.00 4.00 1.40 2.10 0.0
```

Number of projectors

Generation of (G)PAW pseudopotential

File `Si.ld1.in`

5

3S	1	0	2.00	0.00	2.00	2.10	0.0
3S	1	0	0.00	6.00	1.40	2.10	0.0
3P	2	1	2.00	0.00	2.00	2.10	0.0
3P	2	1	0.00	6.00	1.40	2.10	0.0
3D	3	2	-2.00	-0.30	2.00	2.00	0.0

Generation parameters

/

&test

/

4

3S	1	0	2.00	0.00	2.00	2.10	0.0
4S	2	0	0.00	0.00	2.00	2.10	0.0
3P	2	1	2.00	0.00	1.40	2.10	0.0
4P	3	1	0.00	4.00	1.40	2.10	0.0

Projectors parameters



The projectors parameters are the same as the generation parameters except for their energy position

Generation of (GI)PAW pseudopotential

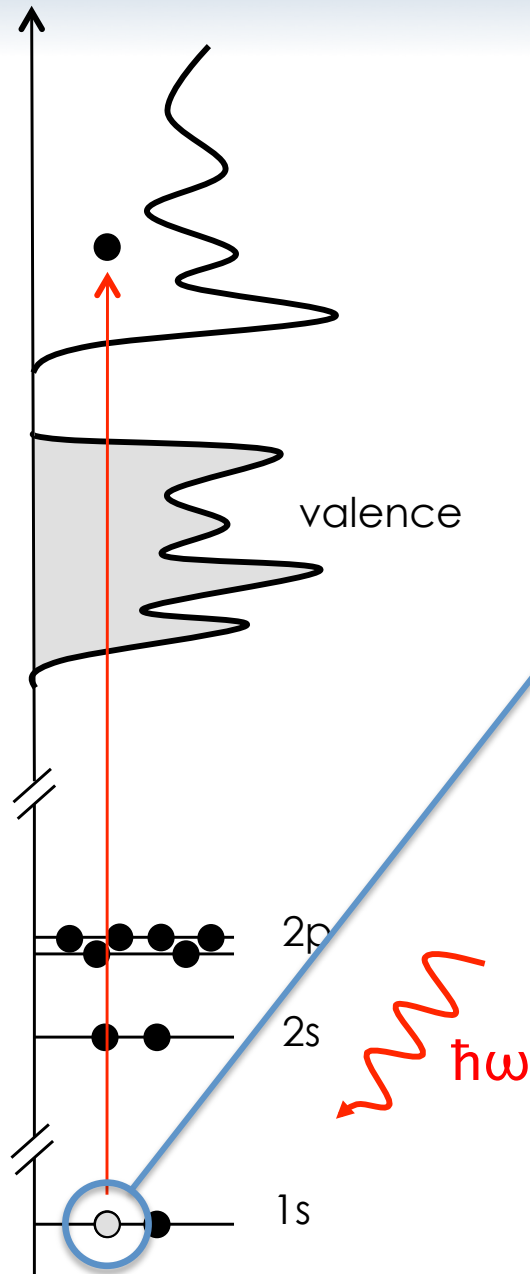
```
$ $BIN/ld1.x < Si.ld1.in > Si.ld1.out
```

Generate the pseudopotential `Si.pbe-us_gipaw.UPF`



Pseudopotentials **must be generated with care.**
Tests are required (but it's not the purpose of this tutorial).

Inclusion of a core-hole



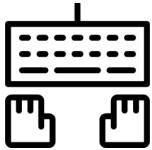
In the final state, a core-hole is present on the absorbing atom

- ➔ It can be included in the pseudopotential
- ➔ In our case of Si-K edge, it corresponds to a 1s core-hole

Including a core-hole in a PP

```
$ cp Si.ld1.in Sih.ld1.in
```

```
$ gedit Sih.ld1.in
```



Changes to make

```
title='Sih'
```

```
config='1s1 2s2 2p6 3s2 3p2 3d-2 4s0 4p0',
```

Hole in the 1s state

```
file_pseudopw='Sih.pbe-us_gipaw.UPF',
```

```
$ $BIN/ld1.x < Sih.ld1.in > Sih.ld1.out
```

Generate the pseudopotential file : Sih.pbe-us_gipaw.UPF

```
$ cp *.UPF ../pseudo
```

```
$ cd ../pseudo
```

Including of a core-hole in a PP

Pseudopotentials including GIPAW informations are available in the online QUANTUM-ESPRESSO pseudopotential table.

<http://www.quantum-espresso.org/pseudopotentials/>

e.g. `Ni.star1s-pbe-sp-mt_gipaw.UPF`

`starNs` : a core-hole in the s state with principal quantum number N.

`PBE` : the exchange-correlation functional used during PP generation

`gipaw` : gipaw information is included in the absorbing atom.

If not, you may find pslibrary on www.qe-forge.org useful :

➡ Add the reconstruction information (gipaw keyword + list of projectors) to the existing `ld1` inputs and generate your own GIPAW pseudopotentials.

Calculation flow for XAS with XSpectra

Prepare the GIPAW pseudopotentials



Extract the core wavefunction



Prepare input file and run SCF calculation



Prepare input file and run XSpectra

Getting core Wave-function

Simply run the following shell script (located in the XSpectra/tools directory in the Quantum Espresso distribution) :

```
$ cd ../pseudo
```

```
$ /scratch/XSpectra/tools/upf2plotcore.sh < Si.pbe-  
mt_gipaw.UPF > Si.wfc
```



The pseudopotential **without a core-hole** should be used : there is no core-hole in the initial state.

The `.wfc` file contains the radial part of $|\psi_i\rangle$ obtained from an atomic AE SCF calculation, it is therefore an AE wave-function. This WF is required by XSpectra.

Calculation flow for XAS with XSpectra

Prepare the GIPAW pseudopotentials



Extract the core wavefunction



Prepare input file and run SCF calculation



Prepare input file and run XSpectra

α - quartz SCF calculation

Trigonal Structure :

- Space Group: $P3_221$

$$a = b = 4.9138\text{\AA},$$

$$c = 5.4052\text{\AA}$$

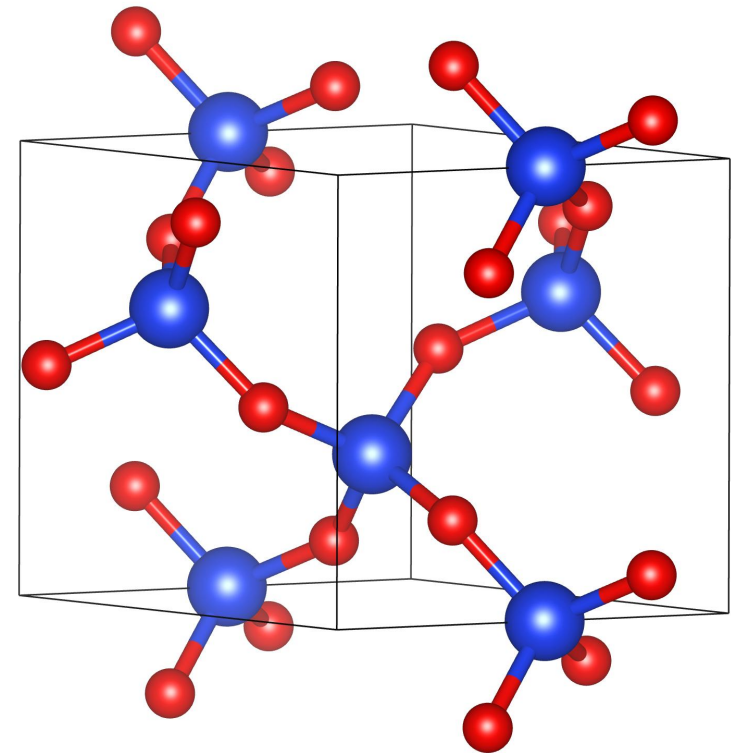
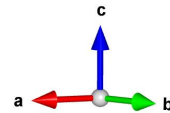
$$\alpha = \beta = 90^\circ,$$

$$\gamma = 120^\circ$$

- 9 atoms in the unit cell :



	X	Y	Z
Si	3a 0.4670	0.0000	0.0000
O	6c 0.4131	0.2677	0.1189



Hexagonal unit cell

α – quartz – SCF calculation

```
$ cd ../SiO2  
$ gedit SiO2.scf.in
```

```
...
```

```
/
```

```
&system
```

```
ibrav = 4 ,
```

```
A = 4.9138,
```

```
C = 5.4052,
```

```
nat = 9 ,
```

```
ntyp = 3 ,
```

```
nspin=1,
```

```
ecutwfc = 40.0,
```

```
ecutrho = 150.0
```

```
/
```

```
...
```

```
/
```

```
ATOMIC_SPECIES
```

```
Si1 28.086 Si.pbe-us_gipaw.UPF
```

```
Si 28.086 Si.pbe-us_gipaw.UPF
```

```
O 15.9994 O.pbe-rrkjus.UPF
```

```
...
```



The absorbing atom needs to be distinguished from the others (even when crystallographically equivalent)

α – quartz – SCF & XSpectra calculation

- Run scf calculation

```
$ $BIN/pw.x < SiO2.scf.in > SiO2.scf.out &
```

➔ Self-consistent calculation of the charge density

- Copy the core wavefunction

```
$ cp ../pseudo/Si.wfc ./
```

- Run XSpectra calculation

```
$ $BIN/xspectra.x < SiO2.xspectra.in > SiO2.xspectra.out &
```

Outline

1. Introduction on XAS
2. The PAW theory
3. Hands on (I) : Si-K edge in α -quartz SiO_2
 - GIPAW pseudopotential generation for Si and $\text{Si}1s^*$
 - SCF calculation for α -quartz SiO_2 unit cell without/with core-hole
 - core wave-function extraction
- 4. The Lanczos algorithm**
5. Hands on (II) : Si-K edge in α -quartz SiO_2
 - XSpectra calculations for α -quartz SiO_2 unit cell without/with core-hole
 - Supercell calculation : $2 \times 2 \times 2$ cell for α -quartz SiO_2
6. Hands on (III) : Dipolar and Quadrupolar contributions at the Ni-K edge in NiO
7. Summary

The sum over the unoccupied states

$$\sigma(\omega) = 4\pi^2 \alpha_0 \hbar \omega \sum_f |\langle \tilde{\psi}_f | \tilde{\varphi}_{\mathbf{R}_0} \rangle|^2 \delta(E_f - E_i - \hbar\omega)$$

The direct, brute-force, sum over the unoccupied states f is **very expensive**. Instead, we use a recursive method proposed originally by Haydock, Heine and Kelly. The cross-section can be re-expressed as :

$$\sigma(\omega) = 4\pi^2 \alpha_0 \hbar \omega \sum_f \langle \tilde{\varphi}_{\mathbf{R}_0} | \tilde{\psi}_f \rangle \delta(E_f - E_i - \hbar\omega) \langle \tilde{\psi}_f | \tilde{\varphi}_{\mathbf{R}_0} \rangle$$

where the imaginary part of the pseudo-Green function $\tilde{G}(E) = (E - \tilde{H} + i\gamma)^{-1}$ associated with the pseudo-Hamiltonian $\tilde{\mathcal{H}} = \mathcal{T}^\dagger \mathcal{H} \mathcal{T}$ (with $E = E_i + \hbar\omega$) :

$$-\frac{1}{\pi} \text{Im} \left[\tilde{G}(E) \right] = \sum_f |\tilde{\psi}_f \rangle \delta(E_f - E) \langle \tilde{\psi}_f |$$

We therefore simply re-write the cross-section under the form of the imaginary part of a matrix element :

$$\sigma(\omega) = -4\pi \alpha_0 \hbar \omega \text{Im} \left[\langle \tilde{\varphi}_{\mathbf{R}_0} | (E - \tilde{H} + i\gamma)^{-1} | \tilde{\varphi}_{\mathbf{R}_0} \rangle \right]$$

R. Haydock, V. Heine and M. Kelly, *J. Phys C* **5**, 2845 (1972)

M. Taillefumier *et al.*, *Phys. Rev. B* **66**, 195107 (2002)

Note : Even under this form (after the sum over f is removed) the numerical task remains very large as the calculation of $\langle \tilde{\varphi}_{\mathbf{R}_0} | (E - \tilde{H} + i\gamma)^{-1} | \tilde{\varphi}_{\mathbf{R}_0} \rangle$ requires a matrix inversion for each energy E .

The Lanczos algorithm and the continued fraction

Matrix element calculated as a continued fraction using the Lanczos algorithm.

➔ The empty states are not calculated explicitly. The sum over empty states depends on the occupied bands only.

1 We use the Lanczos recursive algorithm to bring $\tilde{\mathcal{H}}$ in a **tridiagonal** form. The Lanczos basis is generated by successively applying $\tilde{\mathcal{H}}$:

➔ $|u_0\rangle = |\tilde{\varphi}_{\mathbf{R}_0}\rangle / \sqrt{\langle \tilde{\varphi}_{\mathbf{R}_0} | \tilde{\varphi}_{\mathbf{R}_0} \rangle}$

➔ $b_1|u_1\rangle = \tilde{\mathcal{H}}|u_0\rangle - a_0|u_0\rangle$ with $a_0 = \langle u_0 | \tilde{\mathcal{H}} | u_0 \rangle$

➔ $b_{i+1}|u_{i+1}\rangle = \tilde{\mathcal{H}}|u_i\rangle - a_i|u_i\rangle - b_i|u_{i-1}\rangle$ with $a_i = \langle u_i | \tilde{\mathcal{H}} | u_i \rangle$ and $b_i = \langle u_{i-1} | \tilde{\mathcal{H}} | u_i \rangle = \langle u_i | \tilde{\mathcal{H}} | u_{i-1} \rangle$

Thus, the pseudo-Hamiltonian in the Lanczos basis reads :

$$\tilde{\mathcal{H}} = \begin{pmatrix} a_0 & b_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ b_1 & a_1 & b_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & b_2 & a_2 & b_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & b_3 & a_3 & b_4 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & b_{n-1} & a_{n-1} & b_n \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & b_n & a_n \end{pmatrix}$$

The Lanczos algorithm and the continued fraction

Matrix element calculated as a continued fraction using the Lanczos algorithm.

➔ The empty states are not calculated explicitly. The sum over empty states depends on the occupied bands only.

2 We calculate the matrix element $\langle \tilde{\varphi}_{\mathbf{R}_0} | (E - \tilde{H} + i\gamma)^{-1} | \tilde{\varphi}_{\mathbf{R}_0} \rangle$ as a continued fraction :

$$\sigma(\omega) = -4\pi\alpha_0\hbar\omega\text{Im} \frac{\langle \tilde{\varphi}_{\mathbf{R}_0} | \tilde{\varphi}_{\mathbf{R}_0} \rangle}{a_0 - E - i\gamma - \frac{b_1^2}{a_1 - E - i\gamma - \frac{b_2^2}{a_2 - E - i\gamma - \frac{b_3^2}{\ddots \frac{b_n^2}{a_{n-1} - E - i\gamma - \frac{b_n^2}{a_n - E - i\gamma}}}}}$$

This is what XSpectra calculates : Needs to be converged, i.e. the Lanczos space should be large enough.

C. Gougoussis et al. *Phys. Rev. B* **80**, 075102 (2009)

M. Taillefumier et al. , *Phys. Rev. B* **66**, 195107 (2002)

Outline

1. Introduction on XAS
2. The PAW theory
3. Hands on (I) : Si-K edge in α -quartz SiO_2
 - GIPAW pseudopotential generation for Si and $\text{Si}1s^*$
 - SCF calculation for α -quartz SiO_2 unit cell without/with core-hole
 - core wave-function extraction
4. The Lanczos algorithm
5. Hands on (II) : Si-K edge in α -quartz SiO_2
 - XSpectra calculations for α -quartz SiO_2 unit cell without/with core-hole
 - Supercell calculation : $2 \times 2 \times 2$ cell for α -quartz SiO_2
6. Hands on (III) : Dipolar and Quadrupolar contributions at the Ni-K edge in NiO
7. Summary

Calculation flow for XAS with XSpectra

Prepare the GIPAW pseudopotentials ✓



Extract the core wavefunction ✓



Prepare input file and run SCF calculation ✓



Prepare input file and run XSpectra

α – quartz – XSpectra calculation

```
$ gedit SiO2.xspectra.in
```

```
&input_xspectra  
  calculation='xanes_dipole'  
  edge='K'  
  prefix='SiO2',  
  outdir='./outdir/',  
  xonly_plot=.false.,  
  xiabs=1,  
  xepsilon(1)=0.0,  
  xepsilon(2)=0.0,  
  xepsilon(3)=1.0,  
  xcoordcrys=.true.  
  x_save_file='SiO2_9at.xanes.sav',  
  xniter=2000,  
  xcheck_conv=50,  
  xerror=0.001,
```

```
/
```

```
...
```

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra  
  calculation='xanes_dipole'  
  edge='K'  
  prefix='SiO2',  
  outdir='./outdir/',  
  xonly_plot=.false.,  
  xiabs=1,  
  xepsilon(1)=0.0,  
  xepsilon(2)=0.0,  
  xepsilon(3)=1.0,  
  xcoordcrys=.true.  
  x_save_file='SiO2_9at.xanes.sav',  
  xniter=2000,  
  xcheck_conv=50,  
  xerror=0.001,
```

/ Type of calculation: xanes_dipole $\mathcal{O} = \hat{\epsilon} \cdot \mathbf{r}$
... xanes_quadropole $\mathcal{O} = \frac{1}{2} (\hat{\epsilon} \cdot \mathbf{r}) (\mathbf{k} \cdot \mathbf{r})$

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra  
  calculation='xanes_dipole'  
  edge='K'  
  prefix='SiO2',  
  outdir='./outdir/',  
  xonly_plot=.false.,  
  xiabs=1,  
  xepsilon(1)=0.0,  
  xepsilon(2)=0.0,  
  xepsilon(3)=1.0,  
  xcoordcrys=.true.  
  x_save_file='SiO2_9at.xanes.sav',  
  xniter=2000,  
  xcheck_conv=50,  
  xerror=0.001,
```

/

...

Type of edge : K, L₂, L₃ or L₂₃

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra
  calculation='xanes_dipole'
  edge='K'
  prefix='SiO2',
  outdir='./outdir/',
  xonly_plot=.false.,
  xiabs=1,
  xepsilon(1)=0.0,
  xepsilon(2)=0.0,
  xepsilon(3)=1.0,
  xcoordcrys=.true.
  x_save_file='SiO2_9at.xanes.sav',
  xniter=2000,
  xcheck_conv=50,
  xerror=0.001,
```

/

...



Prefix and outdir of the scf calculation

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra
  calculation='xanes_dipole'
  edge='K'
  prefix='SiO2',
  outdir='./outdir/',
  xonly_plot=.false.,
  xiabs=1,
  xepsilon(1)=0.0,
  xepsilon(2)=0.0,
  xepsilon(3)=1.0,
  xcoordcrys=.true.
  x_save_file='SiO2_9at.xanes.sav',
  xniter=2000,
  xcheck_conv=50,
  xerror=0.001,
```

/

...



Rank of the absorbing atom in the `ATOMIC_SPECIES` list of the scf

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra
  calculation='xanes_dipole'
  edge='K'
  prefix='SiO2',
  outdir='./outdir/',
  xonly_plot=.false.,
  xiabs=1,
  xepsilon(1)=0.0,
  xepsilon(2)=0.0,
  xepsilon(3)=1.0,
  xcoordcrys=.true.
  x_save_file='SiO2_9at.xanes.sav',
  xniter=2000,
  xcheck_conv=50,
  xerror=0.001,
```

/

...

Coordinates of the incident x-ray polarization vector (in crystal coordinates because of xcoordcrys)

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra  
  calculation='xanes_dipole'  
  edge='K'  
  prefix='SiO2',  
  outdir='./outdir/',  
  xonly_plot=.false.,  
  xiabs=1,  
  xepsilon(1)=0.0,  
  xepsilon(2)=0.0,  
  xepsilon(3)=1.0,  
  xcoordcrys=.true.  
  x_save_file='SiO2_9at.xanes.sav',  
  xniter=2000,  
  xcheck_conv=50,  
  xerror=0.001,
```

/

...

Save file storing the Lanczos *a* and *b* parameters

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra
  calculation='xanes_dipole'
  edge='K'
  prefix='SiO2',
  outdir='./outdir/',
  xonly_plot=.false.,
  xiabs=1,
  xepsilon(1)=0.0,
  xepsilon(2)=0.0,
  xepsilon(3)=1.0,
  xcoordcrys=.true.
  x_save_file='SiO2_9at.xanes.sav',
  xniter=2000,
  xcheck_conv=50,
  xerror=0.001,
```

/

...

Maximum number of iterations (maximum dimension of the Lanczos basis)

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra
  calculation='xanes_dipole'
  edge='K'
  prefix='SiO2',
  outdir='./outdir/',
  xonly_plot=.false.,
  xiabs=1,
  xepsilon(1)=0.0,
  xepsilon(2)=0.0,
  xepsilon(3)=1.0,
  xcoordcrys=.true.
  x_save_file='SiO2_9at.xanes.sav',
  xniter=2000,
  xcheck_conv=50,
  xerror=0.001,
```

/

...

Number of iterations between two convergence checks

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&input_xspectra
  calculation='xanes_dipole'
  edge='K'
  prefix='SiO2',
  outdir='./outdir/',
  xonly_plot=.false.,
  xiabs=1,
  xepsilon(1)=0.0,
  xepsilon(2)=0.0,
  xepsilon(3)=1.0,
  xcoordcrys=.true.
  x_save_file='SiO2_9at.xanes.sav',
  xniter=2000,
  xcheck_conv=50,
  xerror=0.001,
```

/

...

Convergence threshold on the integral of the XAS cross-section

α – quartz – XSpectra calculation

file SiO2.xspectra.in

&plot

```
xnepoint=1000,
```

```
xemin=-10.0,
```

```
xemax=50.0,
```

```
gamma_mode='constant'
```

```
xgamma=0.9,
```

```
cut_occ_states=.true.,
```

```
terminator=.true.,
```

/

...

Range in energy for the XANES calculation: [xemin;xemax] with xnepoints

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&plot
  xnepoint=1000,
  xemin=-10.0,
  xemax=50.0,
  gamma_mode='constant'
  xgamma=0.9,
  cut_occ_states=.true.,
  terminator=.true.,
/
...
```

Lorentzian broadening parameters (related to the finite lifetime of the core-hole).

Here, a constant broadening of 0.9 eV over the whole energy range.

➡ Can be made energy-dependant.

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&plot
  xnepoint=1000,
  xemin=-10.0,
  xemax=50.0,
  gamma_mode='constant'
  xgamma=0.9,
  cut_occ_states=.true.,
  terminator=.true.,
/
...
```

Cut smoothly the occupied states (useful for metallic systems)

See Ch. Brouder, M. Alouani, K.H. Bennemann *Phys. Rev. B* **54** 7334 (1996) for details

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&plot  
  xnepoint=1000,  
  xemin=-10.0,  
  xemax=50.0,  
  gamma_mode='constant'  
  xgamma=0.9,  
  cut_occ_states=.true.,  
  terminator=.true.,  
/  
...
```

if `.true.` imposes the use of a terminator, $(a_i; b_i) = (a_N; b_N)$ for $i > N$, allowing an analytical form of the continued fraction.

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&pseudos  
  filecore='Si.wfc',  
/  
  &cut_occ  
    cut_desmooth=0.1,  
/  
4 4 4 0 0 0
```

Core wavefunction file

α – quartz – XSpectra calculation

file SiO2.xspectra.in

```
&pseudos
  filecore='Si.wfc',
/
  &cut_occ
  cut_desmooth=0.1,
/
4 4 4 0 0 0
```



Brillouin zone sampling

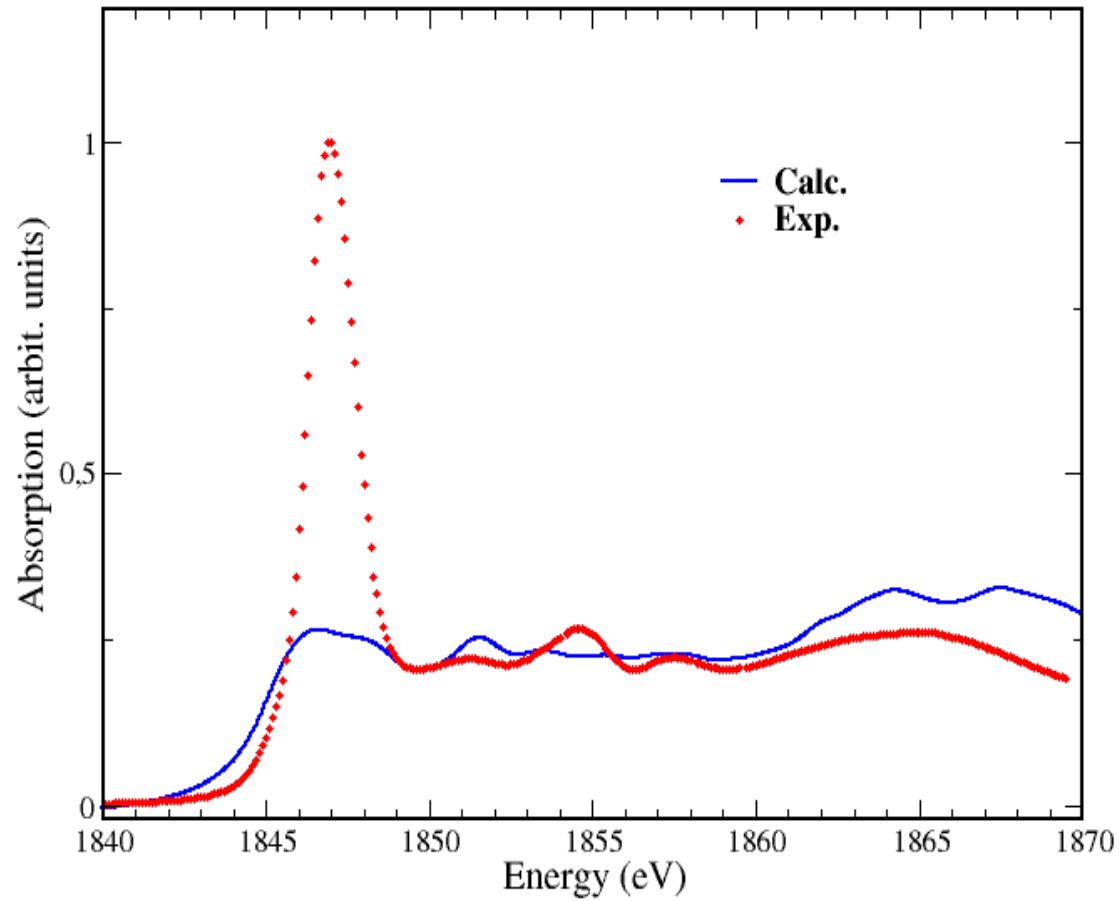
The k-point sampling is not necessarily the same as in the SCF run.

α – quartz – XSpectra calculation

- `SiO2.xspectra.out` contains information about the run
- `xanes.dat` contains the XAS spectrum
(can be visualized with usual plotting tools)
- `SiO2.xanes.sav` save file, containing information on the Lanczos process (a and b vectors)

```
$ mv xanes.dat SiO2.xanes.dat   Rename  
$ gnuplot  
> plot 'SiO2.xanes.dat'
```

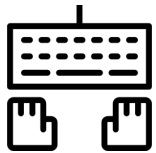
α – quartz – XSpectra calculation



➔ We did not include the core-hole in our calculation !

α – quartz - Calculation with a core-hole – SCF Part

```
$ cd ../SiO2h  
$ mv SiO2.scf.in SiO2h.scf.in Rename  
$ gedit SiO2h.scf.in
```



3 changes to make

1) Change prefix:
prefix='SiO2h'

2) Add in nameliste system:
tot_charge=+1

```
&system  
...  
ecutrho = 150.0  
tot_charge=+1  
/  
...
```

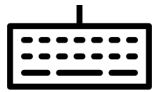
3) Change the name of the pseudopotential file for the absorbing atom:
Si1 28.086 Sih.pbe-us_gipaw.UPF

```
$ $BIN/pw.x < SiO2h.scf.in > SiO2h.scf.out &
```

➡ Run the scf calculation in the presence of a core-hole

α – quartz - Calculation with a core-hole – XSpectra Part

```
$ mv SiO2.xspectra.in SiO2h.xspectra.in Rename  
$ gedit SiO2h.xspectra.in
```



2 changes to make



1) Change prefix:

```
prefix='SiO2h'
```

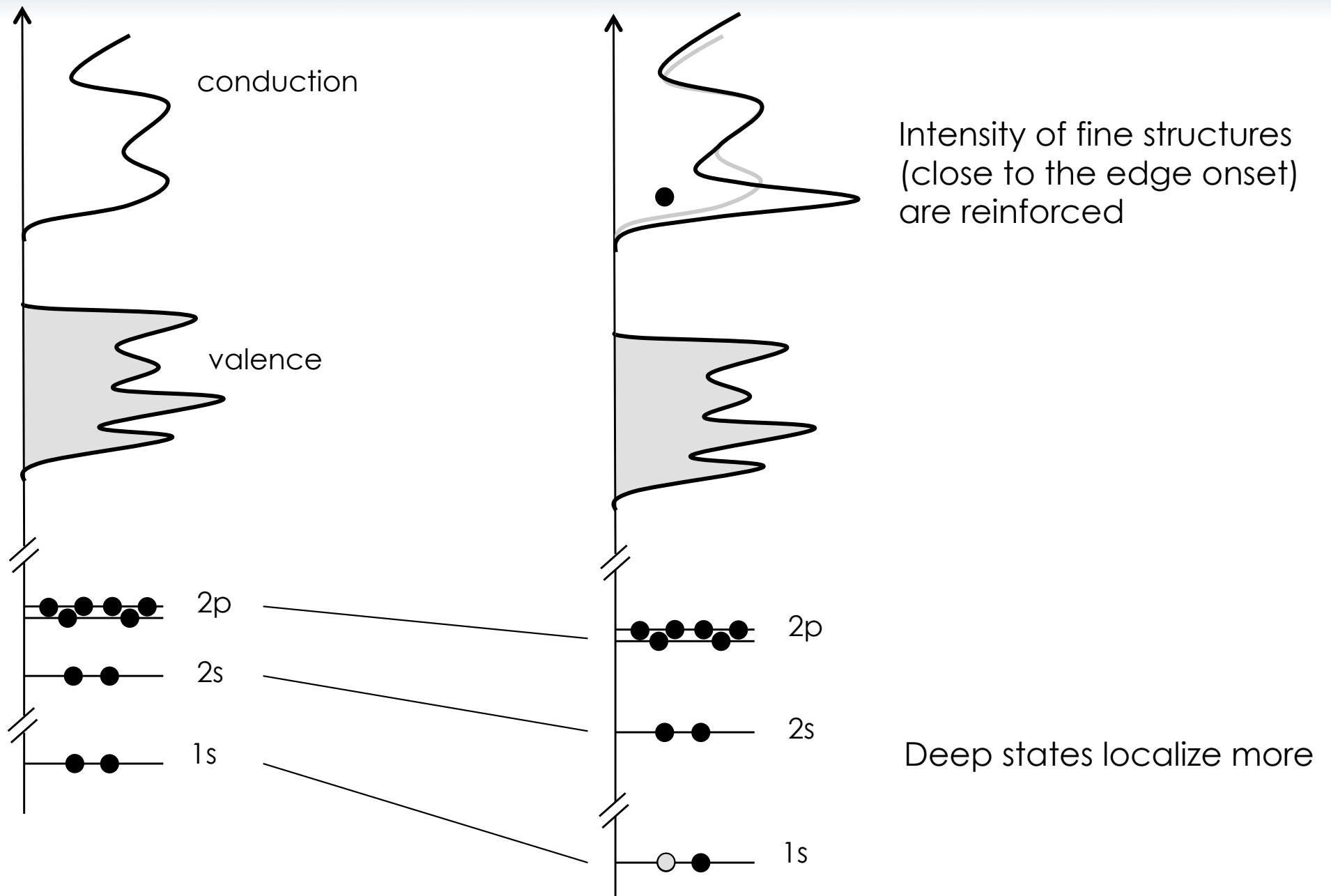
2) Change .sav file name:

```
x_save_file='SiO2h.xanes.sav',!
```

```
$ $BIN/xspectra.x < SiO2h.xspectra.in > SiO2h.xspectra.out
```

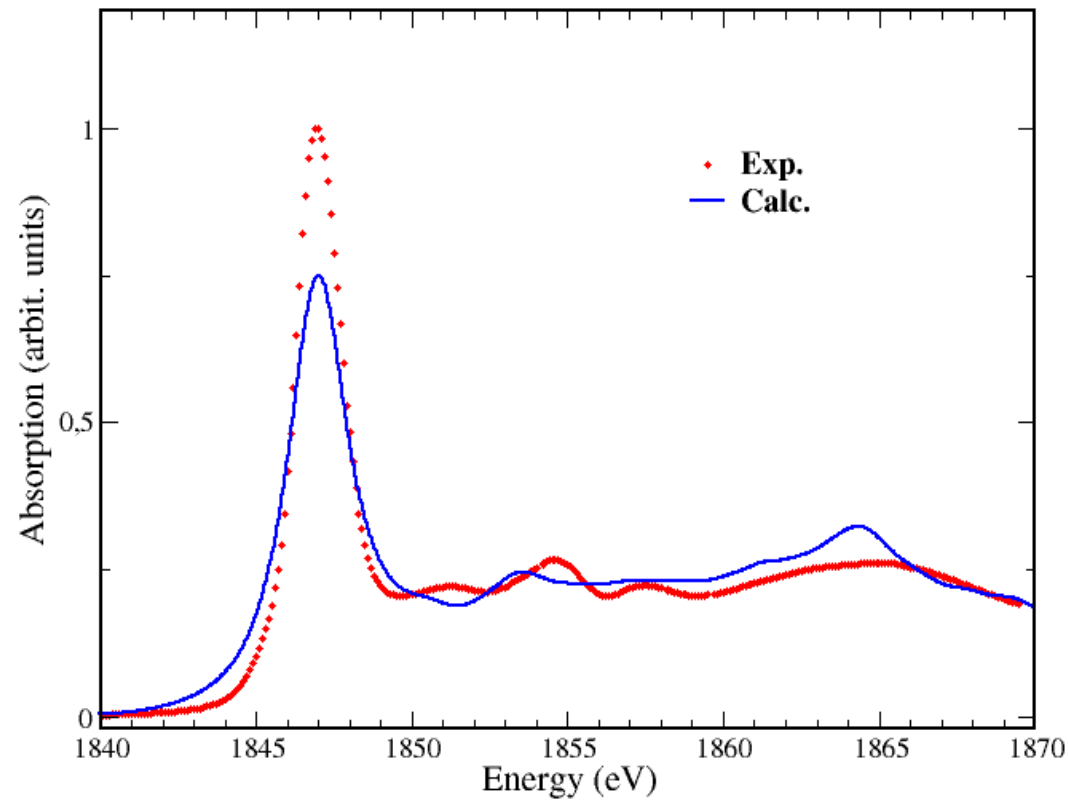
➔ Run the XSpectra calculation in the presence of a core-hole

Quantitative core hole effect (in insulators)



α - quartz - Calculation with a core-hole – XSpectra Part

```
$ mv xanes.dat SiO2h.xanes.dat  
$ gnuplot  
> plot 'SiO2h.xanes.dat'
```

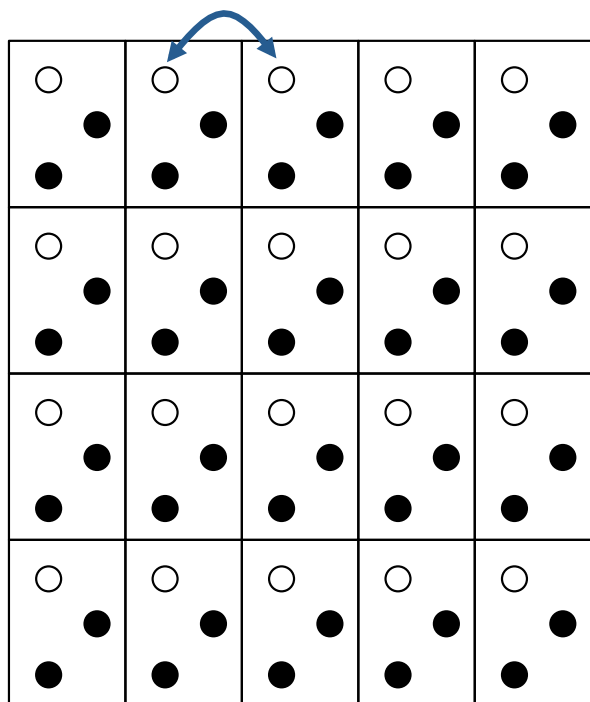


➡ Let's try with a supercell !

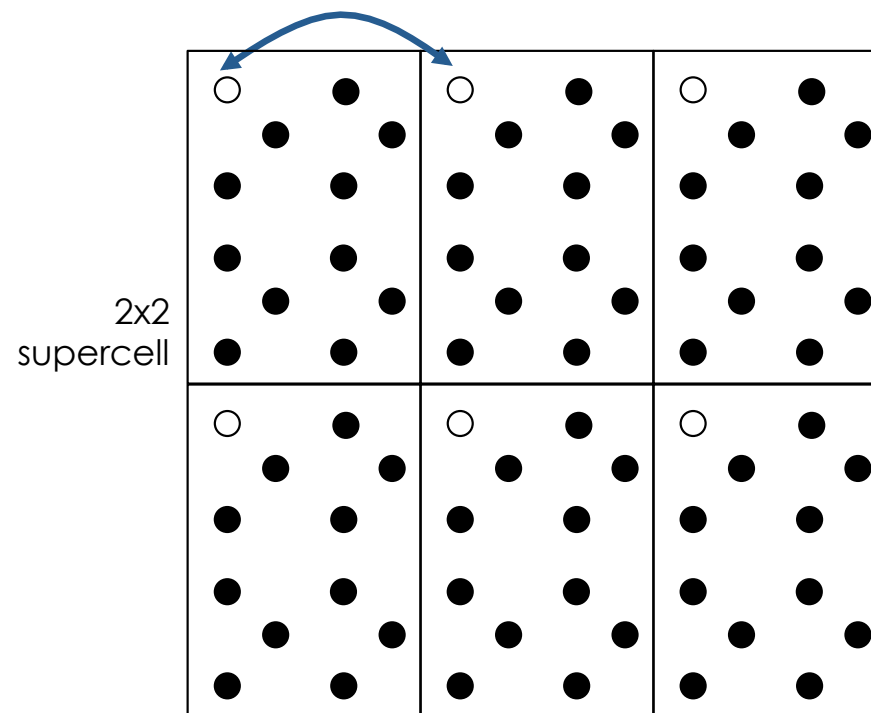
Supercell calculations

Due to periodic boundary conditions, we have to avoid spurious interaction of the excited atom with its periodically repeated images.

Large interaction



Weak interaction



- atom with a core-hole
- atom without a core-hole

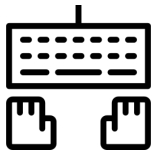
Usually 8 - 10 Å separation is enough

α – quartz – Construction of a 2x2x2 supercell

```
$ cd ../SiO2supercell
$ gedit SiO2_AtomsPositions.txt
$ bash supercell.sh      (home-made script)
  > SiO2_AtomsPositions.txt
  > 2 2 2
```

Create a file supercell.txt with the atomic positions in the supercell

```
$ mv SiO2h.scf.in SiO2_super.scf.in
$ gedit SiO2_super.scf.in
```



1) Remove the atomic positions

```
Si1 0.4700000000000000 0.0000000000000000 0.0000000000000000
Si  0.0000000000000000 0.4700000000000000 0.6666666666666666
Si -0.4700000000000000 -0.4700000000000000 0.3333333333333333
```

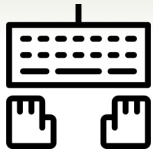
....

2) Copy and paste the atomic positions from `supercell.txt` to `SiO2_super.scf.in`.



Don't forget to rename the absorbing atom Si1

α – quartz – Scf with a supercell



Other changes to make

1) Change prefix:

```
prefix='SiO2_super'
```

2) Change the number of atoms (9 atoms/unit cell x 2x2x2 cells)

```
nat=72
```

3) Change the lattice parameters:

```
A = 9.8276
```

```
B = 10.8104
```

4) Divide the number of k points:

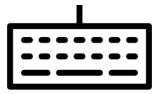
```
K_POINTS automatic
```

```
1 1 1 0 0 0
```

```
$ $BIN/pw.x < SiO2_super.scf.in > SiO2_super.scf.out &
```

α – quartz – Xspectra with a supercell

```
$ mv SiO2h.xspectra.in SiO2_super.xspectra.in  
$ gedit SiO2_super.xspectra.in
```



3 changes to make



1) Change prefix:

```
prefix='SiO2_super'
```

2) Change .sav file name:

```
x_save_file='SiO2_super.xanes.sav',
```

3) Divide the number of k-points:

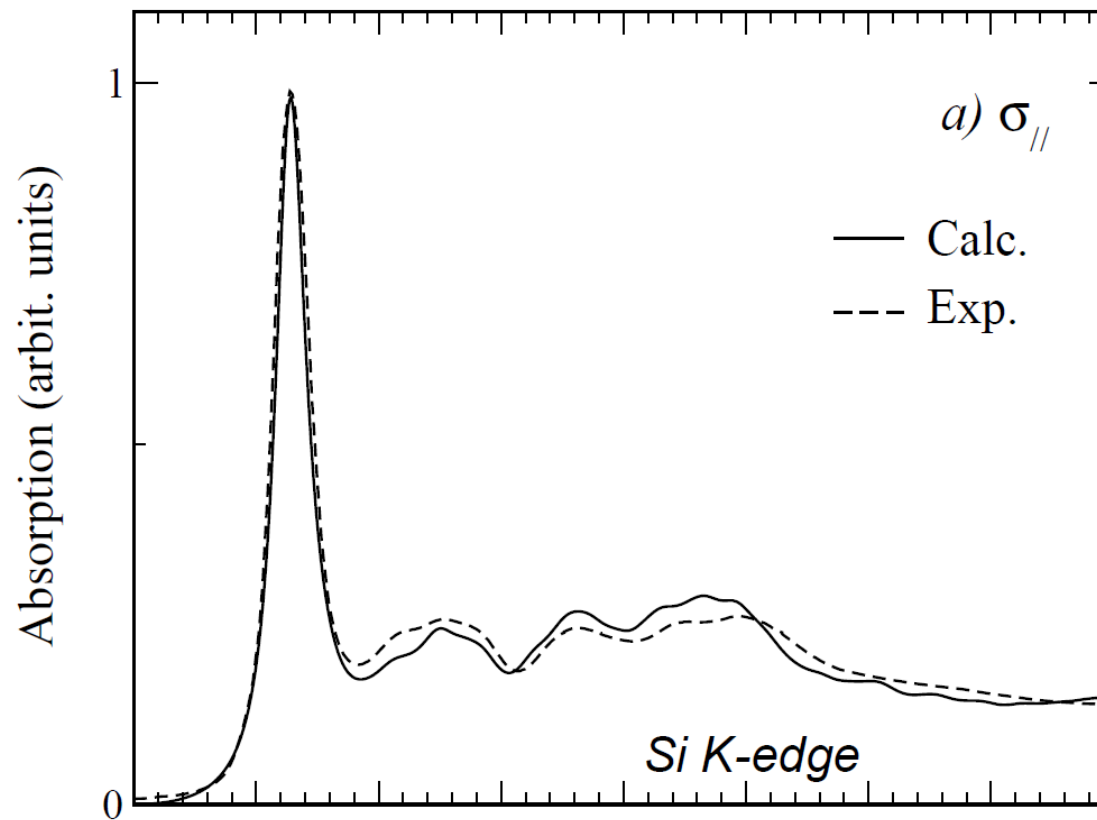
```
2 2 2 0 0 0
```

```
$ $BIN/xspectra.x < SiO2_super.xspectra.in >  
SiO2_super.xspectra.out &
```

α - quartz - Xspectra with a supercell

Results for a polarization $\hat{\epsilon} = (0, 0, 1)$

```
$ mv xanes.dat SiO2_super.xanes.dat  
$ gnuplot  
> plot 'SiO2_super.xanes.dat'
```



XNLD (X-ray Natural Linear Dichroism)

α -quartz has a trigonal space group ($P3_221$, $n^\circ 154$) with an hexagonal unit cell and the point group D_3 :

➔ **it is a dichroic compound !**

The cross-section along the threefold symmetry axis ($\sigma_{||}$) is different from the cross-section in the perpendicular direction (σ_{\perp})

➔ Up to now, we calculated the $\sigma_{||}$ cross-section with $\hat{\epsilon} = (0, 0, 1)$ because the ternary axis is parallel to the **c** axis of the crystal

➔ To calculate σ_{\perp} , any direction perpendicular to **c** is fine, i.e. $\hat{\epsilon} = (1, 0, 0)$

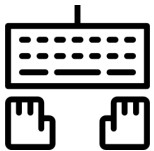
Detailed analysis of the angular dependence of XAS can be found in :

C. Brouder, *J. Phys.: Cond. Mat.* **2** 701-738 (1990)

Angular dependence	Point groups
Isotropy (i)	O_h ($m\bar{3}m$), T_d ($\bar{4}3m$), O (432), T_h ($m\bar{3}$), T (23)
Dichroism (ii)	$D_{\infty h}$ (∞/m), $C_{\infty v}$ (∞m), D_{6h} ($6/mmm$), D_{3h} ($\bar{6}m2$), C_{6v} ($6mm$), D_6 (622), C_{6h} ($6/m$), C_{3h} ($\bar{6}$), C_6 (6), D_{3d} ($\bar{3}m$), C_{3v} ($3m$), D_3 (32), S_6 ($\bar{3}$), C_3 (3), D_{4h} ($4/mmm$), D_{2d} ($\bar{4}2m$), C_{4v} ($4mm$), D_4 (422), C_{4h} ($4/m$), S_4 ($\bar{4}$), C_4 (4)
Trichroism (iii <i>a</i>)	D_2 (222), C_{2v} ($mm2$), D_{2h} (mmm)
Trichroism (iii <i>b</i>)	C_2 (2), C_s (m), C_{2h} ($2/m$)
Trichroism (iii <i>c</i>)	C_1 (1), C_i ($\bar{1}$)

α – quartz – XSpectra with a supercell - perp

```
$ cp SiO2_super.xspectra.in SiO2_super_perp.xspectra.in  
$ gedit SiO2_super_perp.xspectra.in
```



2 changes to make

1) Change xepsilon:

```
xepsilon(1)=1.0,  
xepsilon(2)=0.0,  
xepsilon(3)=0.0,
```



Crystal coordinates because xcoordcrys=.true.

2) Change .sav file name:

```
x_save_file='SiO2_super_perp.xanes.sav',
```

Run the calculation with $\hat{\epsilon} = (1, 0, 0)$

```
$ $BIN/xspectra.x < SiO2_super_perp.xspectra.in >  
SiO2_super_perp.xspectra.out &
```

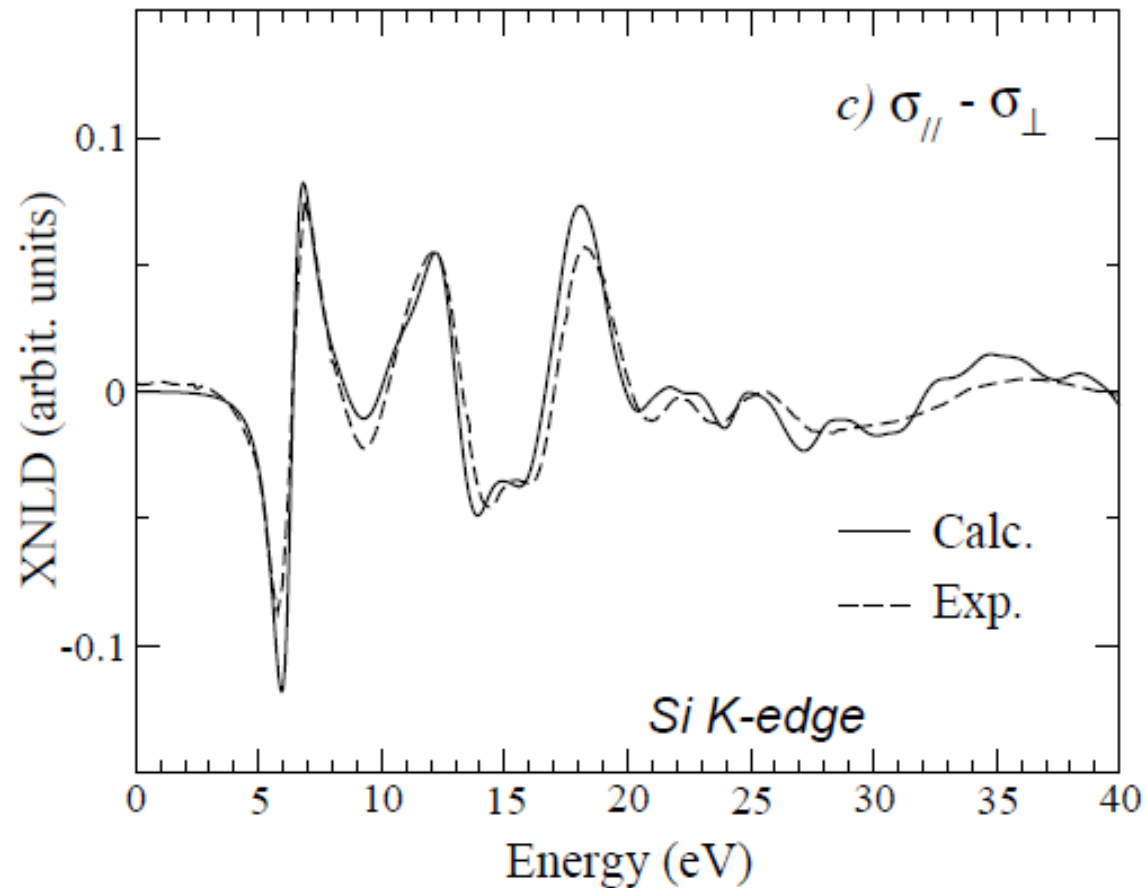

α – quartz – XSpectra with a supercell

Plot XNLD

```
$ mv xanes.dat SiO2_super_perp.xanes.dat
```

```
$ gnuplot XNLD
```

α -quartz dichroic signal @ the Si-K edge :



Calculation flow for XAS with XSpectra

Prepare the GIPAW pseudopotentials ✓



Extract the core wavefunction ✓



Prepare input file and run SCF calculation ✓



Prepare input file and run XSpectra ✓

2nd example:
Ni K-edge in NiO

NiO - SCF calculation

```
$ cd ../NiO
$ gedit NiO.scf.in

...
/
  &system
...
  nspin=2,
  starting_magnetization(1)=1.0,
  starting_magnetization(2)=-1.0,
  tot_magnetization = 0,
...
/
ATOMIC_SPECIES
Ni1 58.6934 Ni_PBE_TM_2pj.UPF
Ni2 58.6934 Ni_PBE_TM_2pj.UPF
O 15.9994 O_PBE_TM.UPF
ATOMIC_POSITIONS crystal
Ni1 0.0000000000 0.0000000000 0.0000000000
Ni2 -.5000000000 1.5000000000 -.5000000000
...
```

NiO - SCF calculation

```
$ cd ../NiO
```

```
$ gedit NiO.scf.in
```

```
...
```

```
/
```

```
&system
```

```
...
```

```
nspin=2,
```

spin-polarized calculation

```
starting_magnetization(1)=1.0,
```

```
starting_magnetization(2)=-1.0,
```

```
tot_magnetization = 0,
```

```
...
```

```
/
```

```
ATOMIC_SPECIES
```

```
Ni1 58.6934 Ni_PBE_TM_2pj.UPF
```

```
Ni2 58.6934 Ni_PBE_TM_2pj.UPF
```

```
O 15.9994 O_PBE_TM.UPF
```

```
ATOMIC_POSITIONS crystal
```

```
Ni1 0.0000000000 0.0000000000 0.0000000000
```

```
Ni2 -.5000000000 1.5000000000 -.5000000000
```

```
...
```

NiO - SCF calculation

```
$ cd ../NiO
```

```
$ gedit NiO.scf.in
```

```
...
```

```
/
```

```
&system
```

```
...
```

```
nspin=2,
```

```
starting_magnetization(1)=1.0,
```

```
starting_magnetization(2)=-1.0,
```

```
tot_magnetization = 0,
```

```
...
```

```
/
```

```
ATOMIC_SPECIES
```

```
Ni1 58.6934 Ni_PBE_TM_2pj.UPF
```

```
Ni2 58.6934 Ni_PBE_TM_2pj.UPF
```

```
O 15.9994 O_PBE_TM.UPF
```

```
ATOMIC_POSITIONS crystal
```

```
Ni1 0.0000000000 0.0000000000 0.0000000000
```

```
Ni2 -.5000000000 1.5000000000 -.5000000000
```

```
...
```

two Ni atoms
antiferromagnetically-coupled

NiO - SCF calculation

```
$ cd ../NiO  
$ gedit NiO.scf.in
```

```
&system
```

```
...
```

```
lda_plus_u=.true.,  
Hubbard_U(1)=7.6,  
Hubbard_U(2)=7.6,
```

DFT + U

Hubbard_U(i): U parameter (eV) for species i

```
...
```

```
/
```

```
ATOMIC_SPECIES
```

```
Ni1 58.6934 Ni_PBE_TM_2pj.UPF
```

```
Ni2 58.6934 Ni_PBE_TM_2pj.UPF
```

```
O 15.9994 O_PBE_TM.UPF
```

```
...
```

NiO – SCF & XSpectra Dipole calculation

- Run scf calculation

```
$ $BIN/pw.x < NiO.scf.in > NiO.scf.out &
```

- Extract the core wavefunction

```
$ cd ../pseudo  
$~/TutorialXSpectra/XSpectra/tools/upf2plotcore.sh <  
Ni_PBE_TM_2pj.UPF > Ni.wfc
```

- Copy the core wavefunction

```
$ cd ../NiO  
$ cp ../pseudo/Ni.wfc ./
```

- Run xspectra calculation

```
$ $BIN/xspectra.x < NiO.xspectra_dip.in >  
NiO.xspectra_dip.out &
```

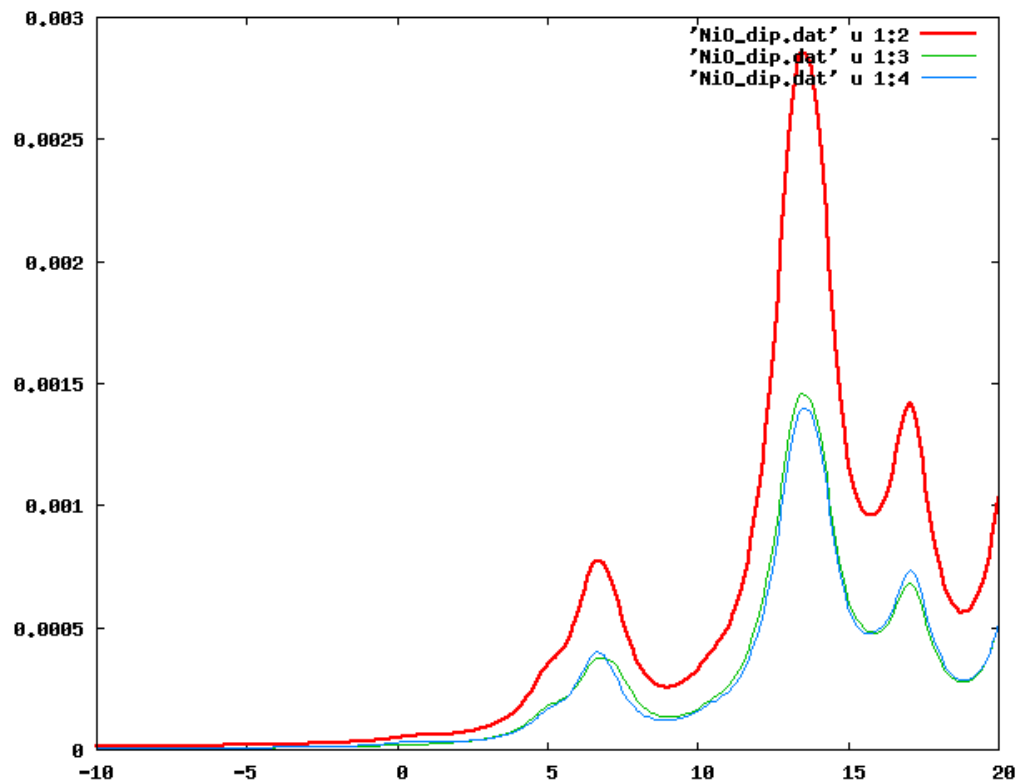

NiO - X Spectra Dipole

```
$ mv xanes.dat NiO.xspectra_dip.dat
```

- Plot up and down contributions to the spectra

```
$ gnuplot
```

```
> plot 'NiO.xanes_dip.dat' u 1:2 w l, 'NiO.xanes_dip.dat' u  
1:3 w l, 'NiO.xanes_dip.dat' u 1:4 w l
```

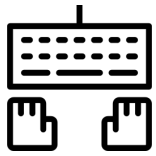


In xanes.dat :

column 1	Total
column 2	Spin up
column 3	Spin down

NiO - XSpectra Quadrupole

```
$ cp NiO.xspectra_dip.in NiO.xspectra_qua.in  
$ gedit NiO.xspectra_qua.in
```



2 changes to make

1) Change calculation :
calculation='xanes_quadrupole',

2) Add xkvec :

xkvec(1) = 1.0,

xkvec(2) = 1.0,

xkvec(3) = -1.0,

wavevector of the incident wave
(required for quadrupole calculations) :

$$\mathbf{k} \perp \boldsymbol{\epsilon}$$

```
$ $BIN/xspectra.x < NiO.xspectra_qua.in > NiO.xspectra_qua.out &
```

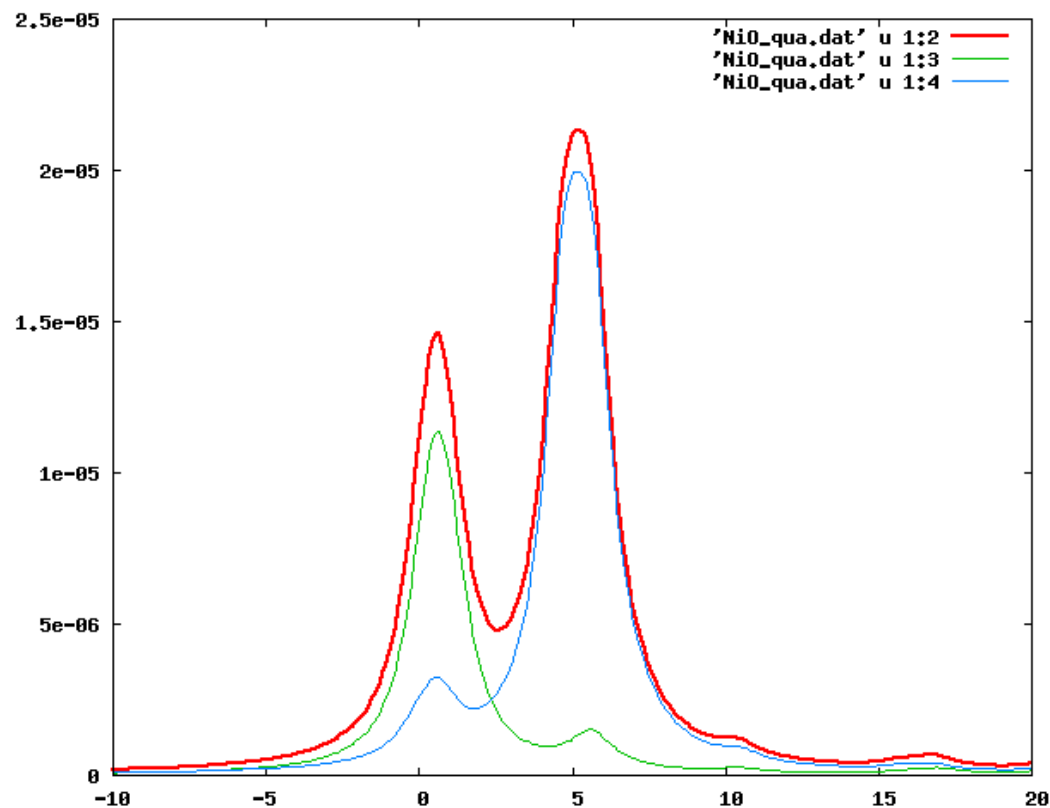
NiO - XSpectra Quadrupole

```
$ mv xanes.dat NiO.xspectra_qua.dat
```

- Plot up and down contributions to the spectra

```
$ gnuplot
```

```
> plot 'NiO.xanes_qua.dat' u 1:2 w l, 'NiO.xanes_qua.dat' u  
1:3 w l, 'NiO.xanes_qua.dat' u 1:4 w l
```



In xanes.dat :

column 1	Total
column 2	Spin up
column 3	Spin down

XSpectra : summary

XSpectra now :

- calculates any **K, L₁, L₂₃ edges** (for which a single-particle description is appropriate !)
- calculates **electric dipole** (K and L edges) and **electric quadrupole** (K and L₁ edges)
- **very efficient** due to the Lanczos algorithm + continued fraction approach
- **supports all standard DFT functionals** available in Quantum Espresso (PZ,PBE,PZ+U,PBE+U)
- supports both **ultrasoft** and **norm conserving** pseudopotentials
- the pseudopotential of the absorbing species must contain information on the core states (GIPAW)
- the *all electron* reconstruction is performed within GIPAW
- a supercell is needed to model the core hole (periodic boundary conditions)

Not yet supported :

- spin-orbit coupling
- circular polarization, XMCD
- hybrid functionals (B3LYP, PBE0, ...)

**Thank you for your attention and good luck
for your calculations !**

Proof : The continued fraction (I)

Assume that we have a Lanczos basis $\{|u_0\rangle, |u_1\rangle, |u_2\rangle, \dots, |u_n\rangle\}$ in which the operator \mathcal{H} has the form

$$\mathcal{H} = \begin{pmatrix} a_0 & b_1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ b_1 & a_1 & b_2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & b_2 & a_2 & b_3 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & b_3 & a_3 & b_4 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & b_{n-1} & a_{n-1} & b_n \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & b_n & a_n \end{pmatrix}$$

We want to calculate the matrix element $\langle u_0 | (\mathcal{H} - \beta \mathcal{I})^{-1} | u_0 \rangle$ with β a scalar

A way to proceed is the following (Gaussian elimination):

1. Use the relations :

$$(\mathcal{H} - \beta \mathcal{I}) |u_n\rangle = (a_n - \beta) |u_n\rangle + b_n |u_{n-1}\rangle$$

$$(\mathcal{H} - \beta \mathcal{I}) |u_{n-1}\rangle = b_n |u_n\rangle + (a_{n-1} - \beta) |u_{n-1}\rangle + b_{n-1} |u_{n-2}\rangle$$

and eliminate the term in $|u_n\rangle$:

$$(\mathcal{H} - \beta \mathcal{I}) \left(|u_{n-1}\rangle - \frac{b_n}{a_n - \beta} |u_n\rangle \right) = \left(a_{n-1} - \beta - \frac{b_n^2}{a_n - \beta} \right) |u_{n-1}\rangle + b_{n-1} |u_{n-2}\rangle$$

Proof : The continued fraction (II)

2. Now, use the previous relation as well as :

$$(\mathcal{H} - \beta\mathcal{I}) |u_{n-2}\rangle = b_{n-1}|u_{n-1}\rangle + (a_{n-2} - \beta)|u_{n-2}\rangle + b_{n-2}|u_{n-3}\rangle$$

to eliminate $|u_{n-1}\rangle$:

$$(\mathcal{H} - \beta\mathcal{I}) \left(|u_{n-2}\rangle - \frac{b_{n-1}}{a_{n-1} - \beta - \frac{b_n^2}{a_n - \beta}} \left(|u_{n-1}\rangle - \frac{b_n}{a_n - \beta} |u_n\rangle \right) \right) = \left(a_{n-2} - \frac{b_{n-1}^2}{a_{n-1} - \beta - \frac{b_n^2}{a_n - \beta}} \right) |u_{n-2}\rangle + b_{n-2}|u_{n-3}\rangle$$

Continuing this way, we get the expression of $|u_0\rangle$ as a function of the $|u_k\rangle$'s :

$$(\mathcal{H} - \beta\mathcal{I}) \left(|u_0\rangle - \sum_{k=1}^n \alpha_k |u_k\rangle \right) = \left(a_0 - \beta - \frac{b_1^2}{a_1 - \beta - \frac{b_2^2}{a_2 - \beta - \frac{b_3^2}{a_3 - \beta - \dots - \frac{b_n^2}{a_{n-1} - \beta - \frac{b_n^2}{a_n - \beta}}}} \right) |u_0\rangle$$

Proof : The continued fraction (III)

Multiplying on the left by $\langle u_0 | (\mathcal{H} - \beta \mathcal{I})^{-1} | u_0 \rangle$, we obtain :

$$\langle u_0 | (\mathcal{H} - \beta \mathcal{I})^{-1} | u_0 \rangle = \frac{1}{a_0 - \beta - \frac{b_1^2}{a_1 - \beta - \frac{b_2^2}{a_2 - \beta - \frac{b_3^2}{\ddots \frac{b_n^2}{a_{n-1} - \beta - \frac{b_n^2}{a_n - \beta}}}}}}$$

The matrix element finally appears as a continued fraction !