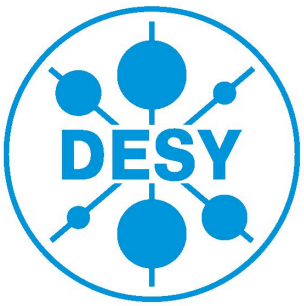

pyFDMNES – a python interface for FDMNES calculations

Tina Weigel, Technische Universität Bergakademie Freiberg, Germany

Supervisor: Dmitri Novikov and Carsten Richter

September 4, 2013

Abstract

The program pyFDMNES is written in python. It is a program for simplified usage of the absorption simulation software FDMNES, that calculates X-ray absorption spectra in the energy range near the absorption edge.
With pyFDMNES it is possible to create the input file for the calculation with data from a CIF file or by an existent input file and to complete or set parameters self-contained. Afterwards the FDMNES calculation is started and data can be fetched from the output file after the simulation with FDMNES. These data can be plotted.

Contents

1. Introduction

X-ray absorption spectroscopy (XAS) is a method for material characterization, to analyze crystals, molecules, surfaces and liquids and get information about the local structure, also magnetic and electric properties. In principle a XAS instrument consists of an X-ray source, monochromators, optical elements and a detector. The detector measures the transmitted beam (detector position behind the sample) or the fluorescence intensity [2]. The schematic layout of XAS experiments is illustrated in fig. 1.

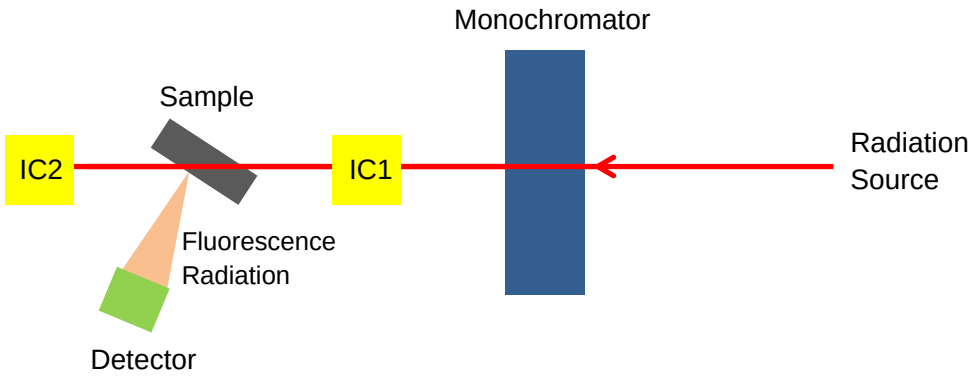

Figure 1: Schematic layout of XAS experiments: the incident photon wavelength is defined by a monochromator and its intensity is measured by ionization chambers (IC1). The radiation penetrates the sample and the transmitted beam intensity is measured (IC2). It is also possible to detect the fluorescence radiation with an energy sensitiv detector [2].

It is possible to use synchrotron radiation as X-ray source. Synchrotron radiation is produced by charged particles that circulate in a storage ring. The particles are deflected in a magnetic field and emit radiation in a wide range including X-rays. So called undulators and wigglers (devices consistion of a structure of magnets with alternating polarization) force the electron beam to small oscillations to produce an intense beam [1]. In absorption spectra the absorption cross section in dependency of the photon-energy is seen. Those spectra can be calculated with simulation software for X-ray absorption spectroscopy. For energies near the absorption edge the calculation tool **FDMNES** is frequently used [5].

The main task of this work was the implementation of a python reference, called **pyFDMNES**. Its purpose is the **FDMNES** application with a simplified input. With **pyFDMNES** it is possible to write the input file, start the **FDMNES** calculation and receive different data from the output file.
Section 2 gives a short overview of the basics of X-ray absorption spectroscopy and the simulations software **FDMNES**. The next section shows the main functions and methods of **FDMNES** (sec. 3) and sec. 4 gives some examples of applications with **pyFDMNES**.

2. Basic informations

2.1. X-ray absorbtion spectroscopy (XAS)

X-ray absorption spectroscopy uses the photoelectric effect for structure characterization. An arriving photon give their completely energy to an electron near the core and leave the atom. A hole in the shell of atom is remaining near the core. The photoelectric effect reduces the intensity of the beam - which is measurable. The absorption of radiation in matter is depending on the atom number and the density. The hole in the near-core shell is filled by an electron from an outer shell. In this process radiation (fluorescence radiation) or a second photelectron (Auger electron) is emitted. This can be used to obtain absorption curves, too [1]. In fig. 2 the principle of absorption is presented.

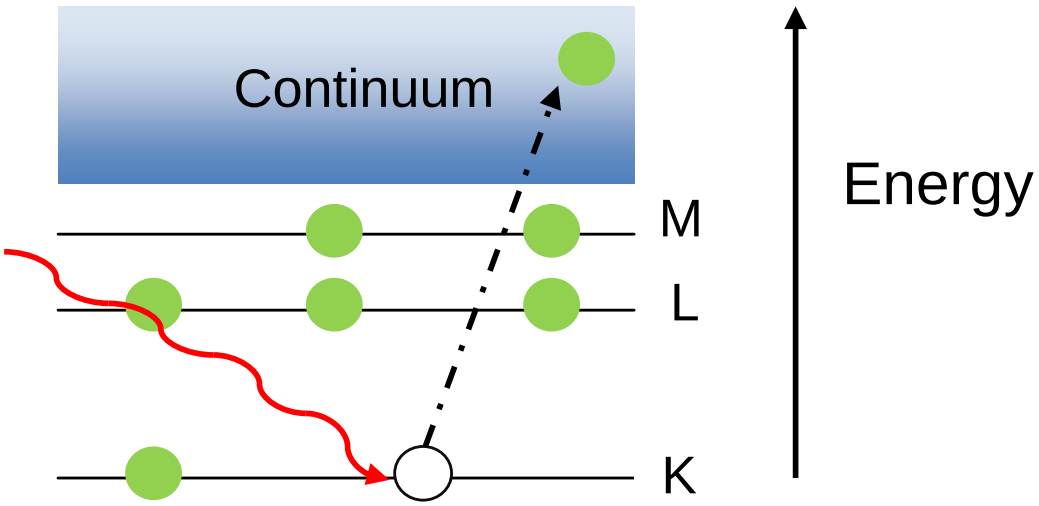

Figure 2: Principle of absorption: the incident radiation (red) is absorped by an electron (green) in the core shell. The electron absorbs the energy of the photon and is emit to free states (contiuum) [2].

The absorption can be discribed by the absorption cofficient. These depends on the absorption cross section, as follows.

$$\mu = \frac{\rho_m N_A}{M} \sigma_a \tag{1}$$

where N_A is Advogadro's number, ρ_m is mass density and M is the molar mass [2]. The absorption coefficient μ is determined via transmission T and is given by

$$T = \frac{I}{I_0} = e^{-\mu z} \tag{2}$$

where I_0 is the intensity of the incident beam and I the intensity diffracted by the sample and z the thickness of sample [2].

For small amounts of resonant atoms fluorescence measurement is preferred (small absorptions), because these has higher sensitity. In this case is assume, that absorption coefficient is approximately proportional to intensity of the fluorescence: $\mu \propto I_F$.
The absorption cross-section is energy dependent. For element specific energy (see fig. 3 at 0 eV) the cross-section jumps. This is called absorption edge. Electrons are bound with discrete energies. When the energy of the photon is larger than that of the electron, the photon interacts with the atom and removes one electron as spherical wave. The probability for photoelectrical absorption then increases abruptly. For lower energies the removal of an electron is not possible and the cross-section is essentially lower [1, 2].

2.1.1. NEXAFS and EXAFS

The absorption cross section is dependent on a variety of physical processes and interaction of the photo-electron with the neighboring atoms. The outgoing spherical electron wave and the back scattered wave interfere. This causes oscillations in the spectra above the absorption edge, called the X-ray absorption fine structure (XAFS). XAFS can be grouped in the near edge X-ray absorption fine structure (NEXAFS) and the extended X-ray absorption fine structure (EXAFS) [2]. In fig. 3, you see the classification of XAFS. NEXAFS or XANES (X-ray absorption near edge structure) is in the energy

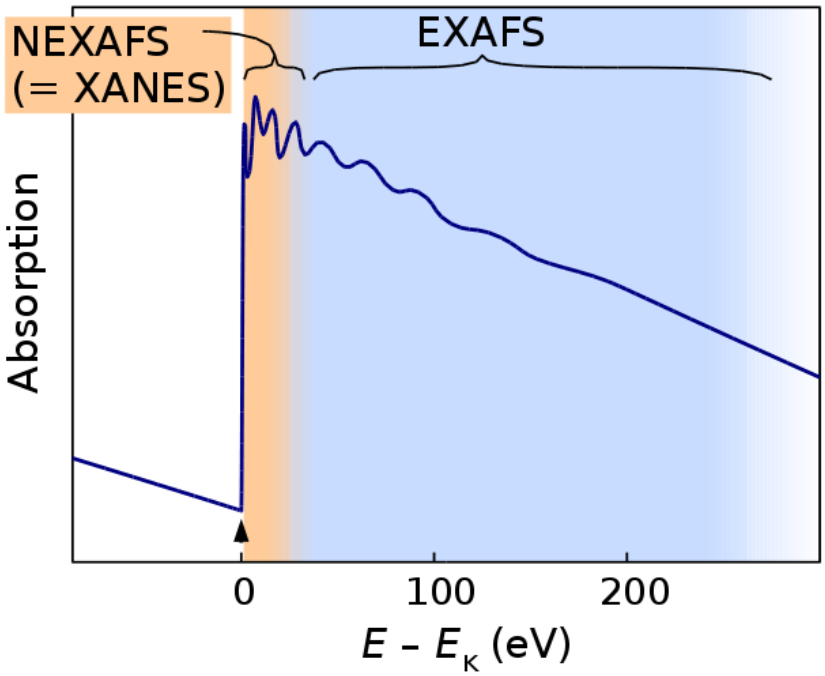

Figure 3: Classification of XAFS [6].

range within 10-50 eV of the edge. It is determined by density of states effect and electron correlation. Furthermore information about the electronic configuration of the absorbing atom are giving. In the EXAFS range for energies 50-1000 eV above the edge single scattering events dominate and give informations about the the local atomic

structure around the absorbing atom [1, 2]. In fig. 4 the principle effect of XAFS is sketched.

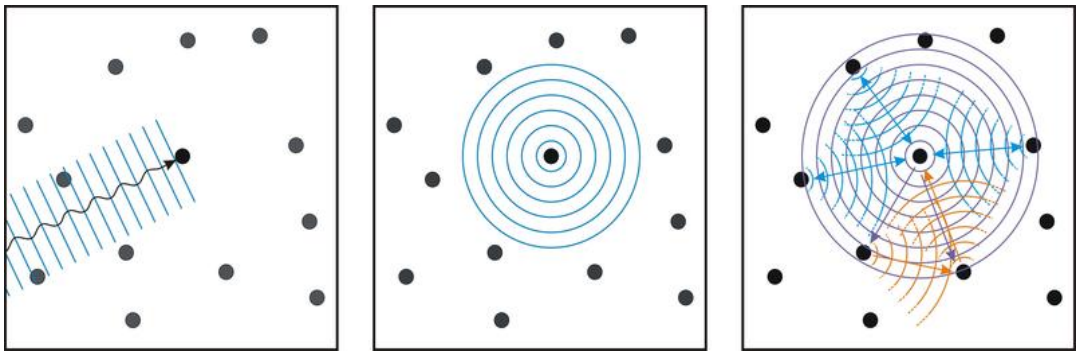

Figure 4: The principle effect of XAFS: in the first picture the incident wave of the photon arrives. After absorption of the photon a spherical electron wave is emitted (second picture). In picture three single scattering events are painted with blue lines and back scattering processes painted with orange lines [7].

2.2. FDMNES – XAS calculation software

FDMNES (finite difference method near edge structure) is a calculation program for x-ray absorption spectroscopy. It calculates the absorption cross section, structure factors and the intensities of anomalous (resonant) diffraction. **FDMNES** provides cartesian and spherical tensors, that are used for the peak-discription of XAS [3].
Parameters for simulation are defined in an input file. **FDMNES** reads this file and starts the calculation of the XANES and RXS (DAFS). After the calculation the convolution of the data is following. The calculated data can be compared with the experimental spectra [5].
FDMNES takes atoms in local clusters around the absorbing atom into account [5]. For calculation of the absorption signal the calculation of density of states (DOS) and the initial states ψ_i are necessary. Therefore the atomic configuration must be given to calculate the DOS for the electrons in the cluster and the initial states of the absorber. Optionally the "self-consistent field" (SCF) theory can be used that minimizes the energy of the cluster to obtain the electron configurations. To obtain the potential the atomic densities are superposed. In the following, the Poisson equation is solved for the charge density resulting in the Coulomb potential. For this potential the energy-dependent exchange-correlation term has to be considered. In the next step a Schrödinger-like equation (SE) is solved and the final states $\psi_f(r)$ are calculated. Now "Fermi golden rule "

$$W_{fi} \propto \sum_q \left|\langle \psi_f \left|\hat{e}_q\right| \psi_i \rangle\right|^2 \delta_{E_f - Ei - \hbar\omega} \tag{3}$$

is applied to get the matrix elements of the section cross section, which are used for the calculation of the absorption coefficient [3]. There are two different methods to calculate the final states. On the one hand the finite difference method (FDM), that uses the full

potential and discretizing the SE. On the other hand the multiple scattering theory (MST) using a constant, spherical and symmetrical crystal potential. The radial SE is solved in each sphere [3].
Finally, the convolution integrates over all unoccupied states and causes a broadening at the spectra to calculate the final quantities. For this the following equal is used [3].

$$\sigma^{conv}(\omega) = \int_{E_F}^{\infty} dE \sigma^{nonconv}(E) \frac{1}{\pi} \frac{\Gamma_f(\omega)}{\Gamma_f(\omega)^2 + (\hbar\omega - E)^2} \tag{4}$$

$\Gamma_f(\omega)$ is a energy dependent width, which is the sum of width of the core hole and the spectral width.

3. The program pyFDMNES

3.1. Methods of pyFDMNES

In the following the functions and methods of the **pyFDMNES** class are discribed (input parameters in brackets).

`add_atom(label, position)`
The method `add_atom` is used to add the atoms to the structure. The label has to start with the chemical symbol of the element. The second argument *position* is a 3d vector in relative unit cell coordinates.

`load_cif(fname, resonant)`
Load data from a "Crystallographic Information File" (CIF) with **pyFDMNES**. It is the standard text file format for representing crystallographic information and has been adopted by the "International Union of Crystallography". The CIF format is useful for higher comparability and uniformity of the representing of crystallographic structures and information [4].
You need the name of the CIF as input. It is also possible to define the resonant atom species. These atoms have a higher influence on the beam absorption. It is defined it by giving the chemical symbol of the atom as second input. The absorption edge will refer to this atom.
The program **pyFDMNES** reads the space group number, space group name, space group origin, the metric of the crystal system and the atom positions in the crystal from the CIF.

`Filein(fname)`
`Filein` loads calculation parameters of an existing **FDMNES** input file. A necessary argument is the name of the **FDMNES** input file. The method reads the file and gets parameters for the calculation, like of radius, range, atom configuration, crystal or molecule structure, the calculation keywords and also the output path. This an easy way to continue preceding simulations.

`Filout(path)`
The method `Filout` writes a new input file for the **FDMNES** calculation.
The input file must have a general structure for the calculation with **FDMNES**. The main keywords are *Range*, *Radius* and *Crystal* (or *Molecul*). It is also possible to add more parameters for the calculation, for example parameters for absorbing atoms (keyword *Absorber*, kind of atoms, who has a higher influence of absorption), density of states (*Density*) or atomic scattering amplitude (*Atom*).
Filout adds the given structure parameters, parameters of the CIF or the parameters of an existing input file to the new file. More parameters for calculation can be given by the user. For input the name or path of the new file has to be specified.

`do_convolution()`
After succesful XANES simulation the convolution can be performed to write a special convolution file with all parameters which are necessary for that. Therefore the method `do_convolution` can be used.

`FDMNESfile()`
`FDMNESfile` is a method to start the calculation with **FDMNES**. It writes the file "fdmfile.txt" and adds the path of input and convolution files. **FDMNES** needs these files for the simulation. It looks for the output files in the same path like the input files or on a given path.

`retrieve()`
The method `retrieve` checks the existance of the created BAV file. If it was successful **pyFDMNES** prints the sentence "Process was sucessful". Subsequently, reflection intensities and XANES curves can be accessed or convolution performed.

`get_XANES(conv)`
It is also possible to fetch information from the output files. To access the data of the XANES simulation, you can use `get_XANES`. The method reads the output file and returnes the data of energy and absorption cross section of the simulation. You can select the convoluted or non convoluted data.

`get_dafs(Reflex, pol_in, pol_out, azimuth)`
To access the anomalous or resonant diffraction (DAFS/RXS) intensities, the methode `get_dafs` is used. For input the values of reflection indices, orientation of the polarization and azimuthal angle are necessary. The DAFS or RXS data for these values is returned. It is also possible to select the convoluted or non convoluted data. If the chosen input parameters are not given, standard values are used.

`checkvars()`
The method `checkvars` check the data types (float, int, str or bool) of the calculation parameters. It will be called before each simulation.

3.2. The input file

With **pyFDMNES** the input file can be created using CIF or an existent input file and complete this or modify it.
This file has to start with the keyword *Filout* and finish with *End*. *Filout* gives the name of the calculation output files and/or the path. The specifications of the files are without file type, because separate output files for different calculation have different file name extension. For example _conv.txt for the file of convoluted XANES spectra or _scan.txt for the file of azimuthal scan in DAFS calculation. The size of the cluster for calculation of the final states is given with the keyword *Radius*. With the keyword *Range*, the energy range relative to the absorption edge for calculation is defined. The

structural data and symmetry are necessary for the calculation. It is given with the keyword *Crystal* or *Molecule*. The first line below the keyword is the crystal metric containing the cell parameters a, b, c and the angles alpha, beta and gamma. In the following lines the atomic number and the positions of all atoms of the structure are given. If the keyword *Absorber* is not defined the first atom in these lines is the resonant or absorbing atom. In the non magnetic case it is sufficient to give the positions of the non-equivalent atoms. To consider the symmetry the space group name or number (the complete name or number as in the international table with origin) must be given. If a convoluted spectrum is needed, the method `do_convolution()` is essential to write a separate file [5].
The listing 1 shows the code to create an input file for rutile and the corresponding convolution.

```
import pyFDMNES
import matplotlib.pyplot as plt
import numpy as np

sim = pyFDMNES.pyFDMNES("TiO2.cif")

sim.Range = np.array([-50, 0.2, 50., 0.5, 100., 2.0, 200])
sim.radius = 2.0
sim.Rpotmax = 8.50
sim.Density = True
sim.Green = True

sim.checkvars()

sim.Filout("tio2_dafs_py_inp.txt")

sim.FDMNESfile()
sim.retrieve()

sim.do_convolution()

sim.FDMNESfile()
sim.retrieve()
```

Listing 1: Code to create an input file and convolution file for TiO_2 from CIF.

The potential inside the cluster is calculated by superposition. The associated radius is given by *Rpotmax*. The keyword *Density* returns information about the density of states. If the more simple and faster but less exact multiple scattering mode is used, the keyword *Green* is necessary [5].

It is possible to add new parameters to an existing input file. An example is shown in listing 2. Several resonant reflections calculation are appended to this file.

```
import numpy as np
import matplotlib.pyplot as plt
import pyFDMNES
sim = pyFDMNES.pyFDMNES("tio2_dafs_py_inp.txt")

sim.extract = True
sim.Reflex = np.array([
                       [0,0,0, 1,1, 45.],
                       [0,0,0, 1,2, 45.],
                       [0,0,1, 1,1, 45.],
                       [0,0,1, 1,2, 45.],
                       [1,1,1, 1,1, 45.],
                       [1,1,1, 1,2, 45.]
                       ])

sim.Filout("tio2_dafs_py_inp2.txt")
sim.FDMNESfile()
sim.retrieve()

sim.do_convolution()
sim.FDMNESfile()
sim.retrieve()
```

Listing 2: Code to add new parameters to an existing input file for TiO_2.

It is not necessary to write all parameters and keywords again. With the keyword *Extract* and the path of the BAV file some information of the file can taken and does not have run the simulation again.
The convolution can be defined by parameters and keywords. For example, *Efermi* specifies the energy of the Fermi level (standard value -5 eV). If the output spectra starts at lower energies, the keyword *Estart* is used. *Ecent* (standard value 30), *Elarg* (standard value 30) and *Gamma_max* (standard value 15) are further convolution parameters and can be defined [5].

4. Accessing for application

4.1. Plot XANES spectra

It is possible to get the data of the output file with **pyFDMNES**. If the simulated XANES spectra is needed, `get_XANES` gives data for plotting (code in listing 3). The code for the input file is shown in listing 1.

```
data = sim.get_XANES(conv = True)
plt.plot (data[:,0], data[:,1], label = "with Convolution")
data = sim.get_XANES(conv = False)
plt.plot (data[:,0], data[:,1], label = "without Convolution")
plt.title("XANES")
plt.xlabel("Energy")
plt.ylabel("Absorption Cross Section")
plt.legend()
plt.show()
```

Listing 3: Code to plot the XANES data for TiO_2.

Figure 5 shows the convoluted and non convoluted XANES spectra (absorption cross section depending on energy) obtained with the example code shown in listing 3.

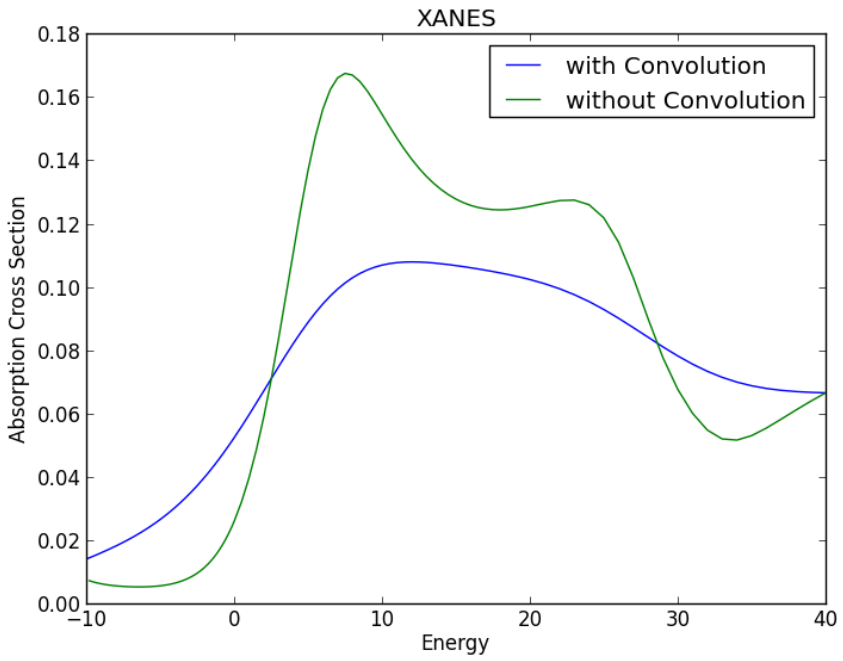

Figure 5: XANES spectra of Rutil: green the non convoluted and blue the convoluted spectrum.

Furthermore a function to change parameters can be script and simulate the spectra for this. In this example the z position of titanium in $BaTiO_3$ is changed in steps of the lattice units. The code for this is shown in listing 4.

```
import numpy as np
import pyFDMNES
import matplotlib.pyplot as plt

sim = pyFDMNES.pyFDMNES("BaTiO3.cif", resonant="Ti")

sim.radius = 2.0
sim.Range = np.array([-10., 0.1, 10, 0.2, 30, 1, 50])
sim.Rpotmax = 8.50
sim.Green = True

sim.checkvars()

zpos = np.linspace(0.3, 0.5, 11) # array of z-positions

plt.title("Convoluted BaTiO$_3$ XANES for different Ti z position")
plt.xlabel("Energy")
plt.ylabel("Absorption Cross Section")

for z in zpos:
    sim.positions["Ti1"][2] = z # set z-coordinate of titanium atom
    sim.Filout("BaTiO3_py_inp.txt")

    sim.FDMNESfile()

    sim.retrieve()
    sim.do_convolution()
    sim.FDMNESfile()
    data = sim.get_XANES(conv=True) # fetch convoluted xanes spectrum

    thislabel = " z = %1.2f"%z
    plt.plot(data[:,0], data[:,1], label = thislabel)

plt.legend(loc = 2)
plt.show()
```

Listing 4: Code to plot the XANES data for $BaTiO_3$ with changing Ti z positions.

In fig. 6 the results can be seen. On certain z positions (deformation of the lattice) pre-edge peaks are visible, which are not visible in non deformed structure.

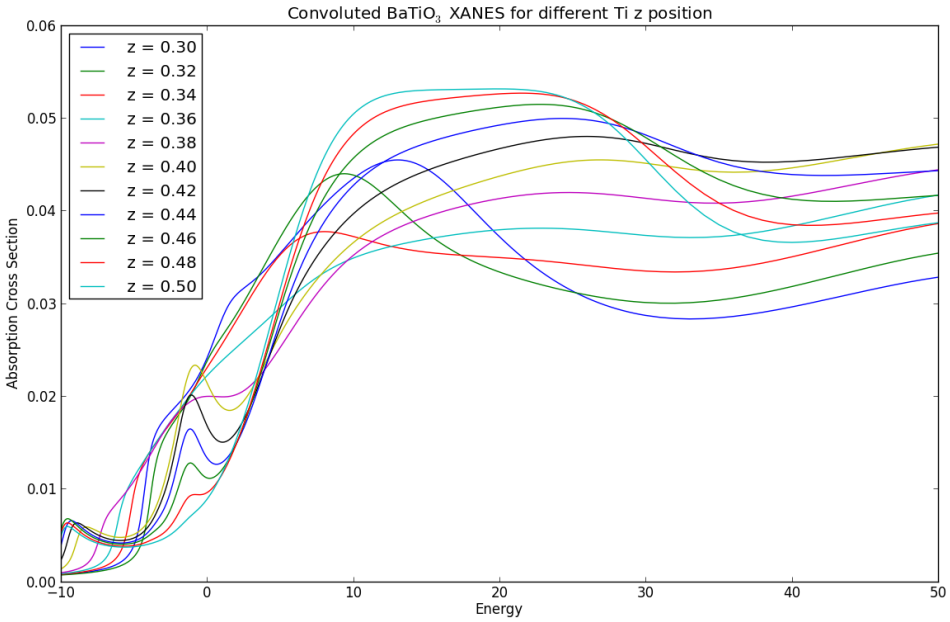

Figure 6: XANES spectra of $BaTiO_3$: the atomic z position of Ti is changed.

4.2. Accessing DAFS

It is shown how to get the data of the DAFS or RXS calculation by `get_dafs` and plot these for defined parameters. Fig. 7 shows the calculated curve for 001 reflection at an azimuthal angle of 45°. The parameters of *RXS* and the code to create an input file can be seen in listing 2. The code to get the convoluted and non convoluted spectra, is shown in listing 5.

```
dafs = sim.get_dafs((0,0,1), 1,1,45., conv=False)
plt.plot (dafs[:,0], dafs[:,(sim.column_real)], label = sim.index_real)
plt.plot (dafs[:,0], dafs[:,(sim.column_im)], label = sim.index_im)
plt.title("RXS %s"%sim.index)
plt.legend(loc = 2)
plt.xlabel("Energy")
plt.ylabel("Amplitude")

dafs = sim.get_dafs((0,0,1), 1,1,45., conv=True)
plt.title("RXS %s"%sim.index)
plt.plot (dafs[:,0], dafs[:,(sim.column)], label = sim.index)
plt.title("RXS %s"%sim.index)
plt.legend(loc = 2)
plt.xlabel("Energy")
plt.ylabel("Intensity")
plt.show()
```

Listing 5: Code to plot the dafs spectra for TiO_2.

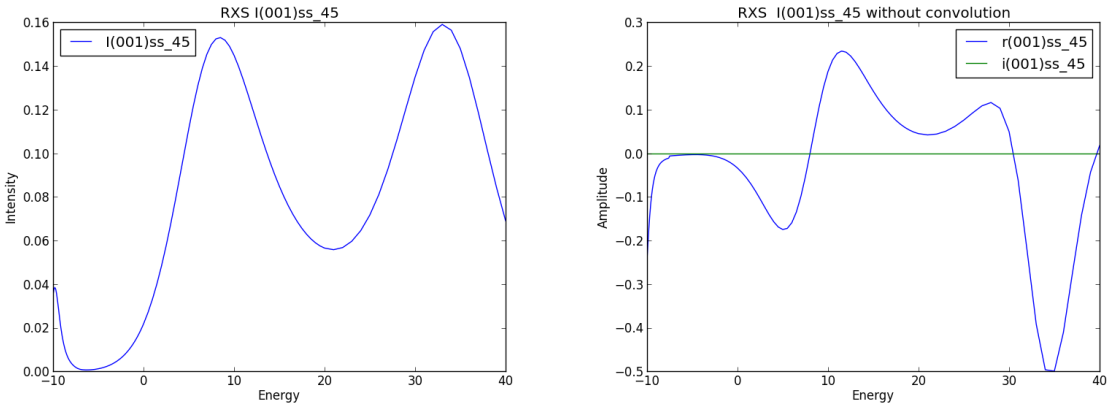

Figure 7: DAFS spectra of the 001 reflex: left the convoluted and right the non convoluted spectra (blue the real and green the imaginary part).

For this example, it is also possible to adds new reflections to the present calculation and plot the data. Here the parameter 002 reflex is added and plotted. The code for that is shown in listing 6. The following code is added to the code in listing 2 which creates an input file.

```
dafs = sim.get_dafs((0,0,2), 1,1,45., conv=False)

sim.Filout("tio2_dafs_py_inp3.txt")
sim.FDMNESfile()
sim.retrieve()

sim.do_convolution()
sim.FDMNESfile()
sim.retrieve()

dafs = sim.get_dafs((0,0,2), 1,1,45., conv=True)
plt.plot (dafs[:,0], dafs[:,(sim.column)], label = sim.index)
plt.legend(loc = 2)
plt.title("RXS %s" %sim.index)
plt.xlabel("Inensity")
plt.ylabel("Absorption Cross Section")

dafs = sim.get_dafs((0,0,2), 1,1,45., conv=False)
plt.plot (dafs[:,0], dafs[:,(sim.column_real)], label = sim.index_real)
plt.plot (dafs[:,0], dafs[:,(sim.column_im)], label = sim.index_im)
plt.legend(loc = 2)
plt.title("RXS %s" %sim.index)
plt.xlabel("Amplitude")
plt.ylabel("Absorption Cross Section")
plt.show()
```

Listing 6: Code to add and plot the dafs spectrum of the additional 002 Reflecion for TiO_2.

Fig. 8 shows the convoluted and non convoluted spectra for the 002 reflection.

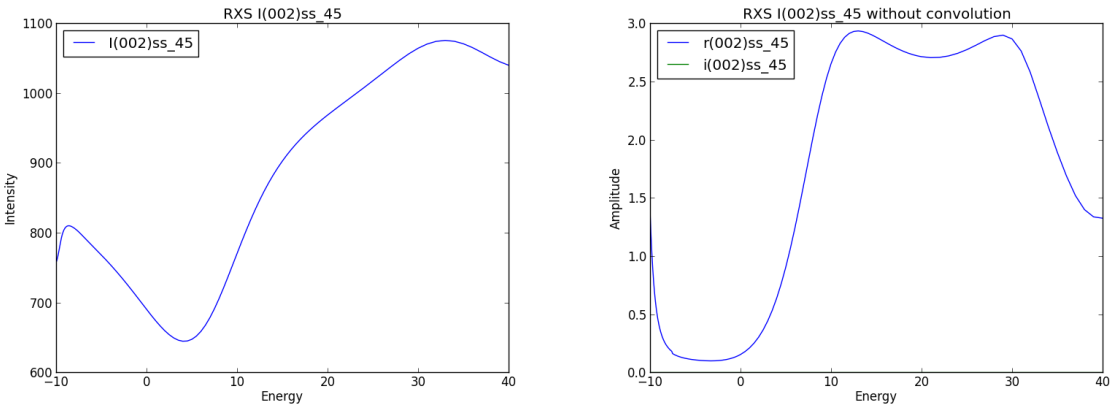

Figure 8: RXS spectra of the 002 reflex: left the convoluted and right the non convoluted spectra.

5. Conclusion and Outlook

The ambition of this work was writing a program for a simplified application of **FDMNES**. This was successful with a python interface called **pyFDMNES**.
In this work the function and methods of **pyFDMNES** are explained and some example of applications are given. It was successful to create the input file and convolution file for the calculation, start the **FDMNES** simulation and process the output files. The absorption cross section data of XANES and DAFS/RXS calculation are easier extracted and plotted in graphics. **pyFDMNES** does all that automatically when using the methods: `get_XANES, get_dafs` and `do_convolusion`.
Scripts can be created to continuously change certain parameters, for example the atom positions at one atom. Example data is calculated and plotted. **pyFDMNES** allows similar scripting and automation for a large number of parameters.
The simple examples just demonstrate the applications of **pyFDMNES** and are far from reality.
This report shows the results after the work of eight weeks. We intent to extent and complete the functions and methods of **pyFDMNES** in future. Foreseen extensions are:

- fit of parameters of low-consumption processes (convolution, fermi levels, edge shift)
- retrieval of arbitary tensor components
- routine and cost evaluation of different parameters (maybe faster convolution)

A. addendum

Figure 9: Attributes and types as well as methods of the **pyFDMNES** class at the current state.

References

[1] Hippert, F., *Neutron and X-ray spectroscopy*, Dordrecht, Springer, 2006

[2] Als-Nielsen, J. and McMorrow, Des., *Elements of modern X-ray physics*, Hoboken, Wiley, edition 2, 2011

[3] Bunău, O. and Joly, Y., *Self-consistent aspects of x-ray absorption calculations*, Journal of Physics: Condensed Matter, volume 21, number 34, 2009

[4] Hall, S. R. and Allen, F. H. and Brown, I. D., *The crystallographic information file (CIF): a new standard archive file for crystallography*, Acta Crystallographica Section A Foundations of Crystallography, volume 47, number 6, pages 655 - 685, 1991

[5] Joly, Yves, *FDMNES User's Guide: Manuel FDMNES*, Institut Néel

[6] Michael Schmid, *http://upload.wikimedia.org/wikipedia/commons/c/c2/NEXAFS_EXAFS_schematic.svg*, 2006, audited 27.08.2013

[7] *http://upload.wikimedia.org/wikipedia/commons/3/34/XAFS.pdf*, 2011, audited 27.08.2013