PROCESSING SPE FILES FROM PRINCETON INSTRUMENTS DURING DATA ACQUISITION IN LNLS

D. B. Beniz†, Brazilian Synchrotron Light Laboratory, Campinas, Brazil

Abstract

DXAS (*Dispersive X-ray Absorption Spectroscopy*) [1] beamline of LNLS (*Brazilian Synchrotron Light Laboratory*) uses a Princeton Instruments CCD, PyLoN [2], to acquire spectra of materials under analysis. Such detector produces an SPE binary file which can be read by a Python script, WinspecUtils.py [3], extracting intensities information on a 2D matrix for each acquired frame and then processing them as a NumPy [4] array.

Using that, a procedure to analyse partial data while the experiment is being performed in DXAS beamline was developed in Python language for their experiments. In this article we will focus on XMCD (*X-ray Magnetic Circular Dichroism*) analysis, describing its motivation and main aspects of its implementation.

MOTIVATION

XMCD experiments in DXAS involve a complex setup of equipment. Characteristics of materials under analysis also contribute to a succeed experiment. The fact is that only after hours of spectra acquisition and more hours of data analysis such results are achieved, and then it could be too late to go back and change something on experiment environment or even in the sample itself.

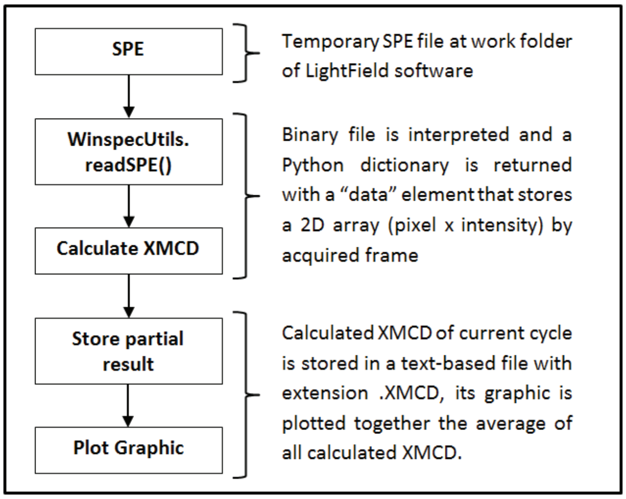
With that in mind, a way to pre-analyse XMCD results during the experiment could save time and effort, helping to make a decision to make something different and maybe remove, or mitigate, injurious interferences on the experiment even before the end of scheduled time of beam usage by the researcher.

The option by Python language to elaborate scripts to achieve such aim was the natural decision in LNLS since it is being recently used in the laboratory at almost all their beamlines to orchestrate the control of devices during experiments, using an internally developed Python package called Py4Syn [5], which is also available for external community. Looking on the Internet by a Python tool that read SPE files, the first step on data analysis, WinspecUtils.py was found and its test was pretty satisfactory. So, we decided to adopt it and develop a procedure to process the data array extracted from the SPE file during the spectra acquisition.

DATA FLOW TO CALCULATE XMCD

Each XMCD cycle is a combination of eight spectra, or frames, which are acquired by PyLoN CCD when the sample were submitted to a magnetic field with a specific sequence of directions, as here: [+ - - + - + + -], where “+” represents the positive direction of magnetic field and “-“ the negative one. To guarantee the synchronization between the applied magnetic field direction and the moment where a spectrum is acquired by the CCD, all the process is controlled by the same script, also in Python, using Py4Syn, which operates the power supplier connected to the magnetic coil and send the electric pulse to trigger the CCD acquisition.

At the end of a cycle, that is, every time a set of eight spectra are acquired, the XMCD is calculated. Figure 1 illustrates the data flow across the main data processing boxes.



Temporary SPE file at work folder of LightField software

Binary file is interpreted and a Python dictionary is returned with a “data” element that stores a 2D array (pixel x intensity) by acquired frame

Calculated XMCD of current cycle is stored in a text-based file with extension .XMCD, its graphic is plotted together the average of all calculated XMCD.

Figure 1: Data Flow to Calculate and Plot XMCD.

WinspecUtils.readSPE()

Based on the well formatted SPE binary file, and Princeton Instruments definition of each type and length of data on its header, readSPE() method of WinspecUtils.py script unpack each piece of data from it and return a Python dictionary with this structure:

```
spedict = {  'data':[],
             'IGAIN': pimaxGain,
             'EXPOSURE': exp_sec,
             'SPEFNAME': spefilename,
             'OBSDATE': date,
             'CHIPTEMP': detectorTemperature,
             'COMMENTS': comments,
             'XCALIB': xcalib,
             'ACCUMULATIONS': accumulations  }
```

The most important to us at this moment is the ‘data’ element. It will store in the first dimension a set of spectra, or frames, and of each spectra a 2D matrix of pixel x intensity.

† douglas.beniz@lnls.br

Maximum number of pixels for PyLoN CCD is 2048, and the intensities for each pixel depends on some variables like beam intensity, beamline monochromator selected energy, distance of CCD to the sample, among others.

Calculate XMCD

With the information of each spectrum of a cycle it is possible to calculate the XMCD. Its formula is presented by Eq. (1), which indicates the difference between the averages of calculated absorbance for each group of acquired spectra, where “NORM+” and “NORM-” indicate, respectively, normalized spectra acquired when magnetic field was positive and when it was negative. Eq. (2) presents the calculation of absorbance (μ), where “I” is the radiant flux transmitted by the material under analysis, and “I0” is the radiant flux it receives.

$$XMCD = \left\langle \mu_{NORM\,+} \right\rangle - \left\langle \mu_{NORM\,-} \right\rangle \qquad (1)$$

$$\mu = \ln\left(\frac{I}{I0}\right) \qquad (2)$$

As the I0 in DXAS beamline can only be acquired by removing the sample from the front of the beam, it needs to be collected before the dichroism experiment, and generated SPE of I0 is one of the inputs for the calculation. WinspecUtils.readSPE() method is also used to gather its data.

NumPy functions are used to facilitate the mathematic calculation of those intermediate and final results manipulating the data directly on arrays.

Storing Partial Results

The calculation results, that is, the XMCD values for all pixels (which are calibrated to correspondent energies in another process that is not presented here), are then stored in an ASCII text file with extension .xmcd. Such file contains information of energy (obtained from pixels and a calibration formula), calculated XMCD, extracted I, extracted I0, calculated μ and calculated μ^{NORM}.

Such files are identified as partial results.

Plotting Graphics

At every cycle processing, with calculation of correspondent XMCD, right after save a file with the partial results, a graphic with current cycle result and an average of all calculated cycles is presented using Matplotlib [6] package of Python. Figure 2 show an example of a graphic for a processed XMCD of an individual cycle and the total average of all previous cycles.

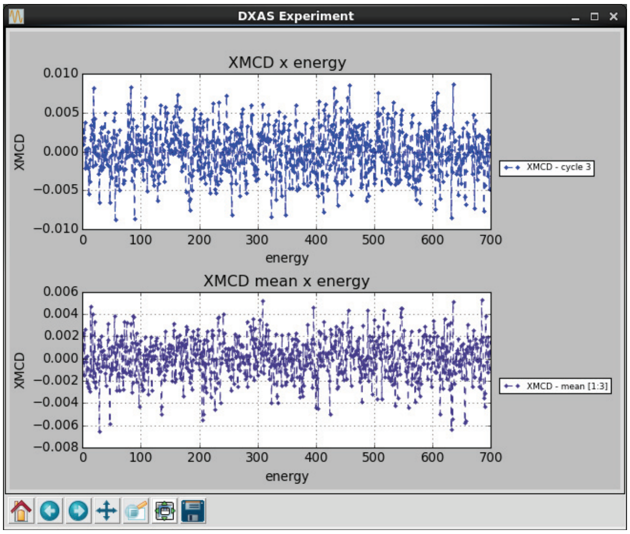


Figure 2: XMCD Graphic of one cycle and all the average.

This way, researchers can follow the behaviour of the material they are analysing and can quickly react if the partial results are totally incoherent with that they presumably should observe.

Finally, at the conclusion of the experiment, a set of ASCII text files is generated with all the calculations, like XMCD and absorbance (μ), for all cycles organized in columns.

REFERENCES

[1] DXAS beamline of LNLS, `http://lnls.cnpem.br/beamlines/xafs/beamlines/dxas`

[2] Princeton Instruments PyLoN, `http://www.princetoninstruments.com/products/PyLoN-CCD`

[3] WinspecUtils.py, `https://github.com/kaseyrussell/python_misc_modules/blob/master/WinspecUtils.py`

[4] NumPy, `http://www.numpy.org`

[5] Py4Syn, `http://py4syn.readthedocs.io/en/latest`

[6] Matplotlib, `http://matplotlib.org`