Processing SPE Files From Princeton Instruments During Data Acquisition in LNLS

Brazilian Synchrotron Light Laboratory

Beniz, D. B., douglas.beniz@lnls.br

CNPEM Brazilian Center for Research in Energy and Materials

DXAS (*Dispersive X-ray Absorption Spectroscopy*) beamline of LNLS uses a Princeton Instruments CCD, PyLoN, to acquire spectra of materials under analysis. Such detector produces an SPE binary file which can be read by a Python script, *WinspecUtils.py*, adapted by Kasey Russell (krussell@post.harvard.edu) from *piUtils.py* module written by James Battat (jbattat@post.harvard.edu). That script extracts intensities information on a 2D matrix for each acquired frame and then process them as a NumPy 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. Here we are simply presenting XMCD (*X-ray Magnetic Circular Dichroism*) analysis, describing its motivation and main aspects of its implementation.

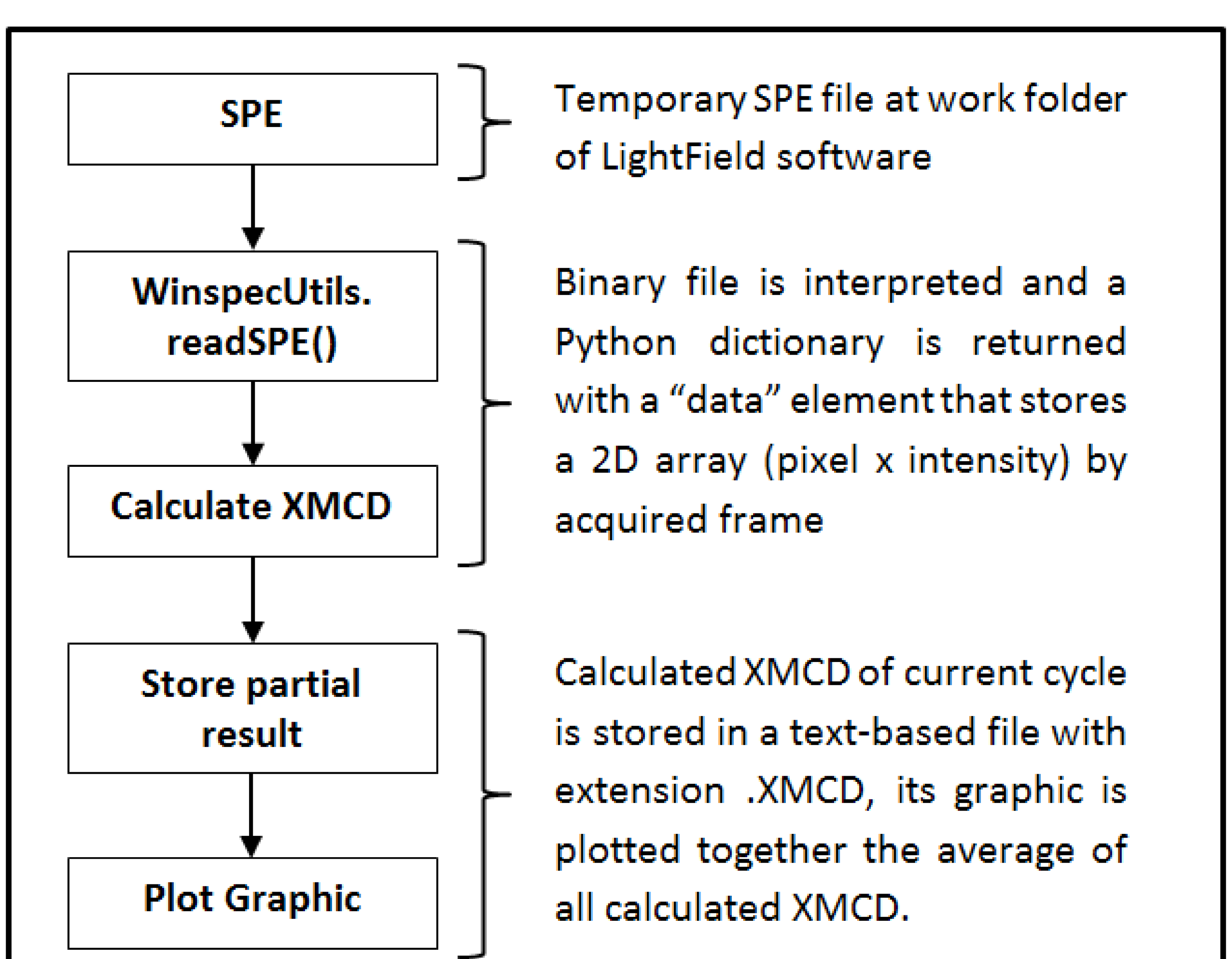
Why to pre-analyse XMCD during experiment?

- 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
- A way to pre-analyse XMCD results during the experiment could save time and effort, helping to take 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.

Why Python?

- Widely used in many different applications, including synchrotron laboratories
- Almost all LNLS beamlines have Python scripts being used to control their operations since we developed Py4Syn package to abstract EPICS (*Experimental Physics and Industrial Control System*) IOC (*Input/Output Controller*) to be used with that language facilities
- Perform mathematical calculations and data matrix manipulation, with NumPy
- And finally, Looking on the Internet for 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



WinspecUtils.readSPE()

```
        # Create a dictionary that holds some header information
        # and contains a placeholder for the image data
        spedict = {'data':[],     # can have more than one image frame per SPE file
                   'IGAIN':pimaxGain,
                   'EXPOSURE':exp_sec,
                   'SPEFNAME':spefilename,
                   'OBSDATE':date,
                   'CHIPTEMP':detectorTemperature,
                   'COMMENTS':comments,
                   'XCALIB':xcalib,
                   'ACCUMULATIONS':accumulations
                   }

```

XMCD Calculation

Each cycle has eight acquired spectra, with magnetic field orientation as: [+ - - + - + + -]

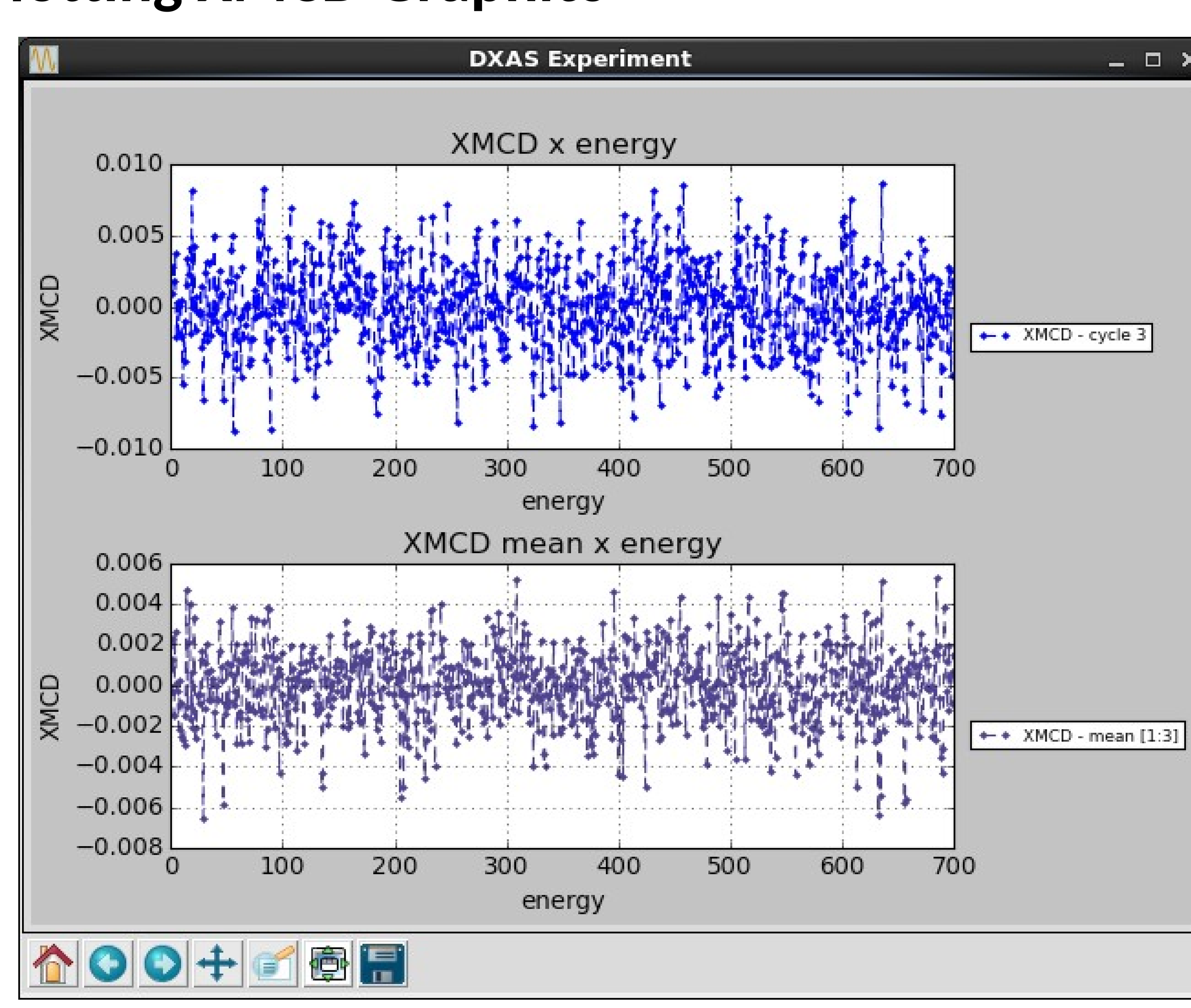
$$XMCD = \langle \mu_{NORM\,+} \rangle - \langle \mu_{NORM\,-} \rangle \qquad (1)$$

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

Stored Data

```
# XMCD - DXAS
#-----------------------------------
#Calibration:
#A:7795.9  B:0.366  C:0.0
#-----------------------------------
#Normalization:
#Eop:7936.0  pr_es:-31.77  pr_ee:-10.31  po_es:7.72  po_ee:25.74  _xflatten:0.0
n_poly:1.0
#-----------------------------------
#energy   xmcd        i       i0       mu         norm
7795.9    -0.3939931  2178.0  21321.0  2.2812855  0.2154347
7796.27   -0.3889252  2189.0  21423.0  2.2810203  0.2250471
7796.63   -0.4213716  2199.0  21445.0  2.2774888  0.2276031
7797.0    -0.3807887  2219.0  21478.0  2.2699725  0.2221603
7797.36   -0.3989592  2238.0  21573.0  2.2658599  0.2235098
7797.73   -0.3949477  2258.0  21654.0  2.2607107  0.2229817
7798.1    -0.3767803  2260.0  21644.0  2.2593635  0.2303476
7798.46   -0.3980429  2258.0  21719.0  2.263708   0.2492561
7798.83   -0.4324432  2276.0  21706.0  2.2551692  0.2416904
7799.19   -0.3922202  2309.0  21690.0  2.2400368  0.22016
7799.56   -0.4036191  2311.0  21752.0  2.2420254  0.2344518
7799.93   -0.4297805  2344.0  21921.0  2.2355862  0.2312454
```

Plotting XMCD Graphics



PCaPAC 2016
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
BRASIL GOVERNO FEDERAL