**Program for XMLD and XMCD spectra analysis**

**Technical documentation**

# Libraries and modules.

## Numpy.

Used for conversion of .txt file containing data to approachable to work with format of data.

## Os.

Used for file handling in the process of uploading and saving.

## Matplotlib.

Used for graphical visualization of data in the form of various plots.

## Re.

Used for searching the file to find individual measurements.

## CSV.

Used for writing results to file in a form enabling further analysis.

## Tkinter.

Used for creating a graphical user interface for the program.

# Uploading.

## Uploading a file.

A standard data set is a .dat file that contains numbers of measurements. Every measurement consists of several columns proceded by several lines of comments describing measurement. Essential for analysis is the first line of comment which contain a number of measurement and its type and two columns of the data.

To load a file *select\_file()*  function is used. It opens File Explorer which allows for choosing a file. If the file is correct the program writes the file line by line to variable *Lines* and closes the file

To find all measurements existing in the whole data set the function *list of measurement()* is used.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Figure 1.** *List of measurement()* function.

There are two regular expressions used:

* pattern\_all – finding all measurements
* pattern\_rscan – finding only measurements made as a function of energy (subject of analysis)

Variable *total* is the number of all measurements in the data set and is used in the program for creating a list of available measurements (measurements are saved in the data set sequentially from the first to the last).

Measurement numbers made as a function of energy are saved in list *rscans*.

## Opening specific measurement.

For opening specific measurements for the purpose of plotting, for example, the function *search in file(nb)* is used.

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

**Figure 2.** Regular expressions in *search in file* function.

There are two regular expressions used (Figure 2.):

* pattern\_first – the first line of commentary of a given measurement (*nb* – number of the measurement you are looking for)
* pattern\_last – the last line of data of a given measurement

To obtain the number of the first line of data of a given measurement, the number of comment lines is added to the number of the first line of commentary (Figure 3.).

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

**Figure 3.** Searching for the first line of data in the *search in file* function.

The function returns a tuple with a number of the first and the last line of data.

# Displaying plots.

## Plotting spectra.

For displaying spectra *Plotter* function is used. The function takes five parameters, including two optional ones:

* *measurement\_nb* – the number of spectra that you want to plot
* *ax* – matplotlib Axes object that you want to use to plot spectrum
* *mode* – the function has two modes of spectra displaying which are activated by passing to the function proper string:

- ‘virgin’: In this mode the raw data is used to draw the graph. This mode is used in the first stages of the program.

- ‘sub and normalized’: In this mode data after normalization and eventual background removal is plotted.

* *w* (optional) – first parameter of fitted background (strip background width)
* *iter\_nb* (optional) – second parameter of fitted background (strip background number of iterations)

The function writes data for specific measurement to a temporary file and extracts two columns from it and save to ndarray *data* (Figure 4.).

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

**Figure 4.** Preparing data for the plotting.

The creation of variable *plot\_obj* allows for getting settings used for plotting.

## Plotting results.

For displaying results the argumentless function *plot\_results* is used.

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

**Figure 5.** Determination of x values.

If the user wants to specify values for the x-axis, the title, and the values will be passed to variables *x\_tittle* and *x\_tab*. They are both strings, so elements in *x\_tab* have to be converted to float numbers. After that, they are saved in the list *x\_int*.

If the user doesn’t want to specify values, *x\_title* is set as ‘x’ and *x\_int* as a list of integers from 0 to the number of the measurements.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

**Figure 6.** Plotting all spectra.

The first plot displayed shows all analyzed spectra. For plotting the function *plotter* is used. Measurement numbers and background parameters are received from the *accepted\_p* dictionary (Figure 6.).

RL for both SROT and ΔRL coefficients are plotted based on results from appropriate lists: *RL3\_0\_tab*, *RL3\_0\_tab*, *dRL3\_tab*.