Neutron ToF Imaging Python Modules Guide

September 2020

Contents

Chapter 1

Introduction

This package includes modules for data reduction and analysis of various Time Of Flight (TOF) neutron imaging data. Initially designed for MCP/Timepix data, most of the modules are compatible with 3D matrixes with dimension (x,y,lambda/TOF). The package includes:

- Jupyter notebooks with data processing/analysis examples (see Chapter 2).
- Tools for Frame overlap Bragg edge imaging (FOBI) data reduction (see Chapter 3).
- Various tools for TOF imaging data processing / visualization (see Chapter 3).
- Advanced Bragg edge fitting of 1d-arrays, but also 2D stack of TOF images (see Chapter 4).
- Bragg edge fitting using derivative's Gaussian of 1d-arrays, but also 2D stack of TOF images (see Chapter 4).
- Fitting of phase fractions, from a linear combination of basis functions (a priori defined phases) (see Chapter 5).

Please find the instruction for the use the functions as comments in the modules or in the chapters of this document.

To use these routines please clone or download the codes to your local machine, then insert the following two lines to your python codes:

```
import sys
sys.path.insert(0, "yourpathtocodes\\ToFImaging\\python$_modules")
```

This may require further installation of external modules.

For questions please contact matteo.busi@psi.ch or anders.kaestner@psi.ch.

Chapter 2

Jupyter Notebooks

Chapter 3

Reduction Tools

3.1 Calibration

3.1.1 tof2l

3.1.2 l2tof

3.1.3 tof2l_t0k

3.2 Image merging

3.2.1 averageimage

3.2.2 medianimage

3.2.3 weightedaverageimage

3.3 Rebinning

3.3.1 binning_ndarray

3.3.2 spectral_binning_resolution

3.3.3 moving_average_1D

3.3.4 moving_average_2D

3.3.5 spatial_image_rebinning

3.3.6 spectral_image_rebinning

3.3.7 savitzky_golay

Chapter 4

Single Bragg Edge Fitting

The modules edgefitting_1D.py and edgefitting_2D.py contain routines for the fitting of a single Bragg edge found in the given pattern or a 2D stack of TOF images.

4.1 edgefitting_1D.py

4.1.1 GaussianBraggEdgeFitting

Performs Bragg edge fitting with gaussian model to an ndarray containing the signal with the length of the spectrum (could be lambda, tof or bin index). INPUT:

- signal = ndarray of the spectrum containing the Bragg edge(s)
- spectrum = spectrum, length of this ndarray must correspond to size of spectrum
- spectrum_range = range corresponding to lambda where to perform the fitting
- est_pos = estimated bragg edge position (in spectrum dimension)
- est_wid = estimated bragg edge width (in spectrum dimension)
- est_h = estimated bragg edge height (in spectrum dimension)
- bool_log = set to True to perform log norm and convert to attenuation
- bool_smooth = set to True to perform Savitzky-Golay filtering of the signal derivative
- smooth_w = window size of S-G filter
- smooth_n = order of S-G filter

- bool_print = set to True to print output

OUTPUT: Dictionary with the following fits in the dimension of the mask

- 'edge_position' : edge position
- 'edge_height': edge height
- 'edge_width': edge width
- 'edge_slope': edge slope
- 'median_image': median Transmission image in the selected lambda range

4.1.2 AdvancedBraggEdgeFitting

Performs Bragg edge fitting with gaussian model to an ndarray containing the signal with the length of the spectrum (could be lambda, tof or bin index). Currently not tested with attenuation data, must be transmission.

INPUT:

- signal = ndarray of the spectrum containing the Bragg edge(s)
- spectrum = spectrum, length of this ndarray must correspond to size of spectrum
- spectrum_range = range corresponding to lambda where to perform the fitting
- est_pos = estimated bragg edge position (in spectrum dimension)
- est_sigma = expected Gaussian broadening
- est_alpha = expected decay constant (moderator property)
- bool_log = set to True to perform log norm and convert to attenuation
- bool_smooth = set to True to perform Savitzky-Golay filtering of the signal derivative
- smooth_w = window size of S-G filter
- smooth_n = order of S-G filter
- bool_print = set to True to print output
- bool_linear = flag to activate linear spectrum assumptions at the sides of the edge (otherwise exponential)

OUTPUT: Dictionary with the following fits in the dimension of the mask

- 't0' = fitted Bragg edge position
- 'sigma' = fitted Gaussian broadening
- 'alpha' = fitted decay constant (moderator property)
- 'a1, a2, a5, a6' = parameters for spectrum besides the edge: a1 and a2 before, a5 and a6 after the edge
- 'final_result' = fitting result after 7th iteration
- 'fitted_data' = final fitted spectrum
- 'pos_extrema' = extrema of the Bragg edges
- 'height' = fitted height of the Bragg edge

4.2 edgefitting_2D.py

4.2.1 GaussianBraggEdgeFitting_2D.py

4.2.2 AdvancedBraggEdgeFitting_2D.py

Chapter 5

Phase Fraction Fitting

Work in progress ...

Chapter 6

Rietveld Fitting

Work in progress ...