SoftwareX 7 (2018) 309–312

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

multiflexxlib: A Python package for data reduction and visualization for the cold-neutron multi energy wide angle analyzer MultiFLEXX

Siqin Meng [a], Rasmus Toft-Petersen [b,c], Lijie Hao [a], Klaus Habicht [c,*]

[a] *China Institute of Atomic Energy, Beijing 102413, China*
[b] *Department of Physics, Technical University of Denmark, DK-2880 kgs. Lyngby, Denmark*
[c] *Helmholtz-Zentrum Berlin für Materialien und Energie, D-14109 Berlin, Germany*

ARTICLE INFO

Article history:
Received 25 June 2018
Received in revised form 20 September 2018
Accepted 20 September 2018

Keywords:
Inelastic neutron scattering
Three-axis spectroscopy
Visualization tool

ABSTRACT

A Python software package for data reduction and visualization of continuous angle multiple energy analysis (CAMEA) type detector backend MultiFLEXX is presented. The software concept focuses on unambiguous, automated aggregation of experimental data and preservation of raw data structure in graphical representation, enabling on-the-fly analysis of experimental data from MultiFLEXX with minimal amount of user input, reducing confusion and human error in studies involving multiple parameters. The software also provides a set of interfaces for versatile tweaking of graphing parameters, facilitating generation of production-quality graphs for use in publications. The software enhances the role of MultiFLEXX as a swift mapping option available at the cold-neutron triple-axis spectrometer FLEXX.

Code metadata

Current code version	1.0.4
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2018_81
Legal Code License	MIT
Code versioning system used	Semantic
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	Python $>$ 2.7 or $>$ 3.5, *numpy, scipy, pandas, matplotlib*
If available Link to developer documentation/manual	See repository location
Support email for questions	mengsq04@gmail.com

1. Introduction and motivation

In material science, the interatomic forces are best investigated using inelastic neutron scattering. Due to the periodicity of the single crystals investigated, these forces give rise to collective excitations of the lattice as a whole. These can be either in the form of waves of moving lattice planes or waves of precessing magnetic moments, where the forces investigated are either the electrostatic repulsion between atoms or the quantum mechanical exchange force between atomic magnetic moments. They generally take the form of quasi-particles, with well defined momenta and energy — where the relationship between momentum (denoted $\mathbf{Q}$) and energy (denoted $\hbar\omega$) defines the so-called dispersion relation. These relations define the dynamics of the material in question, and are investigated when neutrons create or annihilate such quasiparticles — losing or gaining momentum and energy in the process of inelastic scattering. In some systems, such as disordered magnetic systems [1,2], it is often interesting to study such dispersion relation over a large portion of $(\mathbf{Q}, \omega)$ parameter space, referred to as a mapping study. Mapping studies can be very time-consuming, due to the large amount of data points to be gathered, and the generally low count rate in INS experiments. Traditionally, triple-axis spectroscopy (TAS) collects data at only one $(\mathbf{Q}, \omega)$ coordinate at a time. It is thus advantageous to improve data acquisition efficiency of mapping studies by collecting scattered neutrons into

[*] Correspondence to: Hahn-Meitner-Platz 1, 14109 Berlin, Germany.
E-mail address: habicht@helmholtz-berlin.de (K. Habicht).

https://doi.org/10.1016/j.softx.2018.09.006
2352-7110/© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

multiple scattering angles (2θ) and with multiple final energies for each angular channel simultaneously. Such concept is known as Continuous Angle Multiple Energy Analysis (CAMEA) [3,4].

The new MultiFLEXX [5,6] backend built for the cold-neutron TAS (cTAS) FLEXX [5,7] at the BER II neutron source, HZB collects neutrons in 31 2θ angular channels that are separated by 2.5°, and 5 fixed energies from 2.5 meV to 4.5 meV on each 2θ channel. The MultiFLEXX backend is dominantly operated in constant-energy mode, where 2-D constant-energy slices of the scattering function $S(\mathbf{Q}, \omega)$ are generated. While MultiFLEXX greatly improves the mapping capability of cTAS FLEXX, the utilization of MultiFLEXX was met with multiple challenges: Experiments on MultiFLEXX often involve multiple parameters such as temperature and magnetic field. The amount of variables can quickly grow and become difficult to manage during an experiment. The difficulty is compounded by the fact that MultiFLEXX experiments often involve multiple measurement passes that partially interleave or repeat each other, possibly with varying step sizes. Additionally, the fixed angular offset between angular channels on MultiFLEXX means that the measurement is carried out using a set of points in $(\mathbf{Q}, \omega)$ space that do not form an orthogonal grid. On MultiFLEXX, due to the relatively small number of data points, further reducing such data by binning into an orthogonal grid of bins can give rise to large errors in $(\mathbf{Q}, \omega)$ coordinates and artifacts by having a small number of data points or even zero data points per bin. Such concern is largely alleviated in time-of-flight measurements due to much larger amount of data points, which is the main focus of modern software development such as Mantid [8] and Horace [9]. In order to address aforementioned tasks, *multiflexxlib* is designed to be a highly automated data reduction and visualization software that enables users and instrument responsible to correctly and quickly manage and analyze data from MultiFLEXX.

2. Software design

Fig. 1 shows the data aggregation pipeline of *multiflexxlib*. Raw data is first parsed into *Scan* objects. *Scan* objects add a layer of abstraction over raw data by providing a set of interfaces for accessing experiment metadata, generating the UB-matrix [10] if not explicitly provided, applying the UB-matrix to data points and keeping track of detector characteristics such as normalization factors. While *multiflexxlib* currently only accept scan files from MultiFLEXX as data source, the *Scan* class can be further abstracted into a Python abstract base class to support a common interface for other CAMEA-type backends. While users can directly access *Scan* objects to perform basic diagnostics and exporting of data if necessary, *Scan* objects are mainly intended to be fed into data aggregation routine and combined into *BinnedData* objects. The aggregation routine identifies points as being identical or different in experiment conditions and angles through a process called binning. The bins are defined through a series of bin edges, any values that fall between two subsequent bin edges are considered identical in the data aggregation process. The binning process first sorts a relevant physical quantity such as incoming neutron energy in ascending order, and subsequently iterate through the sorted values, adding a bin edge whenever two values are different by more than the instrument control precision tolerance. This way, bin edges are created based on proximity of values, avoiding the potential pitfall of splitting values that are intended to be identical into two bins by having a pre-defined bin edge too close to a nominal value, and relieving users of the burden of debugging such scenarios and manually keeping track of bin edges. A bigger tolerance can be passed into the aggregation routine if needed. Alternatively, a smaller tolerance can be used if a focused study is performed with step length smaller than usual tolerance. The binning routine can also be operated in a mode using a regular grid of bins if preferred. The aggregation routine first performs binning of datasets based on experiment conditions including initial energy (E_i), temperature and magnetic field, and subsequently aggregates data from identical conditions based on 2θ angles, sample rotation angles ($A3$) and final energies (E_f). The user is only required to explicitly specify if the binning tolerance is to be overridden from instrument defaults. This way, data aggregation is performed with a high degree of robustness and automation. *multiflexxlib* makes extensive use of the *pandas* [11] library in data aggregation, especially its ability to embed arbitrary data types directly in its *DataFrame* data structure. The *matplotlib* [12] library is used for graphics generation in this package.

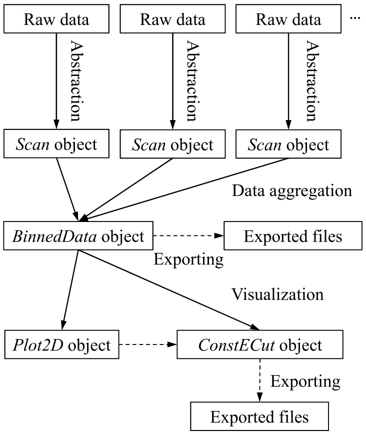

Fig. 1. Data aggregation pipeline.

The *BinnedData* object contains aggregated data from *Scan* objects derived from individual scan runs. *BinnedData* objects generate 2-D constant energy plots and 1-D subsets (referred to as "cuts") by spawning the corresponding objects. The *BinnedData* object keeps track of measurement parameters, and is the main point of interaction for interactive and scripted use. Care is taken to design an interface that is both easy to use and versatile enough for advanced needs: the most basic usage of generating 2-D colormap plot from raw data only contains 3 lines of code with no required parameters in function calls. If there is further need for customization, *matplotlib* objects can be accessed through provided properties to enable the entire range of customization possible with *matplotlib* package. The aggregated data and 1D-cuts can be exported into comma-separated values (CSV) format files if required.

3. Software functionalities and illustrative examples

3.1. Constant-energy mapping

Constant energy maps of magnetic excitations in the antiferromagnet MnF_2 [13,14] generated using *multiflexxlib* are shown in Fig. 2. A common strategy of adapting non-orthogonal scatter of data points to 2-D colormaps is to perform interpolation over input data. While such technique is adequate for a sufficiently fine mesh of input data, measured points of MultiFLEXX can be sparse compared to the size of features of the measured excitation spectra depending on scan parameters. Fig. 2a and b show graphs generated by performing linear interpolation over the non-orthogonal grid of measured data points. It can be seen that the data point density information is obscured in this case. To address this issue, *multiflexxlib* performs 2-D Voronoi partition [15] of input data points in reciprocal space in the scattering plane. Alternatively, the Voronoi partition can be performed in angles and subsequently converted to reciprocal coordinates. A similar