



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

Meng, Siqin; Toft-Petersen, Rasmus; Hao, Lijie; Habicht, Klaus

Published in:
SoftwareX

Link to article, DOI:
[10.1016/j.softx.2018.09.006](https://doi.org/10.1016/j.softx.2018.09.006)

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Meng, S., Toft-Petersen, R., Hao, L., & Habicht, K. (2018). multiflexlib: A Python package for data reduction and visualization for the cold-neutron multi energy wide angle analyzer MultiFLEXX. *SoftwareX*, 7, 309-312. <https://doi.org/10.1016/j.softx.2018.09.006>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



multiflexlib: 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.

© 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/>).

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, <i>numpy</i> , <i>scipy</i> , <i>pandas</i> , <i>matplotlib</i>
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}, ω) 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}, ω) 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).

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 MultiFLEX [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 MultiFLEX 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 MultiFLEX greatly improves the mapping capability of cTAS FLEXX, the utilization of MultiFLEX was met with multiple challenges: Experiments on MultiFLEX 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 MultiFLEX 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 MultiFLEX means that the measurement is carried out using a set of points in (\mathbf{Q}, ω) space that do not form an orthogonal grid. On MultiFLEX, 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}, ω) 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, *multiflexlib* 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 MultiFLEX.

2. Software design

Fig. 1 shows the data aggregation pipeline of *multiflexlib*. 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 *multiflexlib* currently only accept scan files from MultiFLEX 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

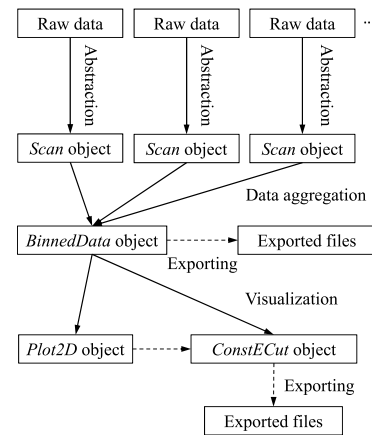


Fig. 1. Data aggregation pipeline.

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 (A_3) 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. *multiflexlib* 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.

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 *multiflexlib* 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 MultiFLEX 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, *multiflexlib* 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

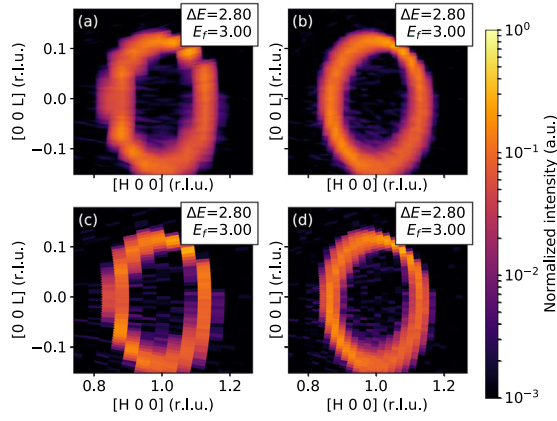


Fig. 2. Constant energy maps of magnetic excitations in MnF_2 measured with MultiFLEX with $E_i=5.8$ meV and $E_f=3.0$ meV. (a) and (c) are generated from one measurement pass; (b) and (d) are generated from two interleaving measurement passes with a 1.25° offset in 2θ angles. (a) and (b) are plotted using bilinear interpolation; (c) and (d) are plotted using the Voronoi tessellation method.

approach is adopted by *multiplot* [16] for the FlatCone [17] multi-analyzer at ILL. This way, each measured point is represented as a discrete region of Q-space that is closer to the corresponding data point than to surrounding points. The users are thus provided with an intuitive and unambiguous representation of raw data structure and measured point density. The simplicity of the principle also ensures that the software is versatile and robust when processing scans that are partially overlapping, partially repeating, done with variable step intervals or contain non-working detector tubes. It is worth mentioning that such method is essentially identical to a nearest neighbor interpolation over an infinitely fine grid, but performing Voronoi partition has the advantage of not requiring an extremely fine interpolated mesh grid and can be zoomed in without causing pixelation. *multiflexlib* also supports handling non-orthogonal axes such as hexagonal a – b plane. The creation of axes and 2-D plots has its own public interface to facilitate the integration into the users' own graph generation routines or other software packages.

3.2. Constant-energy 1-D cut and dispersion relation plot

It is also useful to extract 1-D cuts on measured data. When performed on experimental data from ToF measurements, such cuts usually involve binning data into a set of bins, typically with multiple measured data points per bin due to the large number of available measured points [9]. When treating MultiFLEX data, due to the lower data point density, setting lateral bin sizes becomes tricky as an overly small setting leads to an overly small number of data points accepted in cut generation, requiring users to tweak bin settings carefully. *multiflexlib* provides an alternative cutting method that operates on the Voronoi partition of measured data. The method draws a line segment between the specified cut start and end points, and each data point corresponding to the crossed Voronoi regions is subsequently projected onto the cut axis. Fig. 3 shows 1-D cuts generated using *multiflexlib*. It can be seen that the algorithm automatically takes advantage of a higher data point density when available without user intervention, and gives a representation of the number of data points involved in the representation of a feature in the excitation, which both are useful in on-the-fly analysis. The user is also relieved of the burden of manually keeping track of and setting lateral tolerance, as it is adapted to the size of the Voronoi tessellation cells. Manual setting

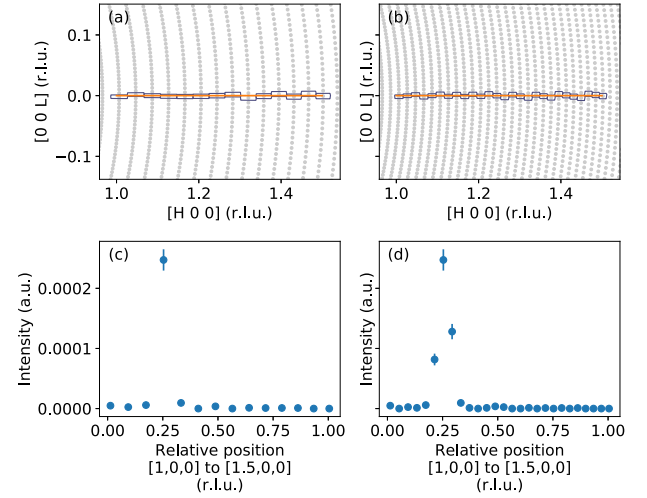


Fig. 3. 1-D const-E cuts of MnF_2 excitations from (1 0 0) to (1.5 0 0) (r.l.u.) with $E_i=5.8$ meV and $E_f=3.0$ meV. The left column is generated from one scan pass, (a) shows the Voronoi cells of the data points involved in cut. (c) shows cut results. The right column shows results generated from two interleaving scan passes with 1.25° offset in 2θ angles.

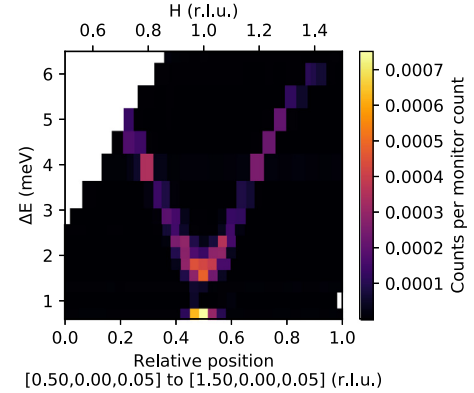


Fig. 4. Dispersion relation of MnF_2 from (0.5 0 0.05) to (1.5 0 0.05) (r.l.u.) measured with MultiFLEX with $E_i=5.1$ meV, 5.8 meV, 8.5 meV and 8.9 meV. Intensity at $\Delta E = 0.6$ meV is from a spurion arising from type-A accidental Bragg scattering [18, p. 148].

of rectangular bin sizes is also supported in a separate function, which is more suited for analysis with sufficient data point density, where multiple points per bin are available. Cuts can be made along any direction for both methods. For both cutting methods, an *inspect* method that overplots bin edges on data points as is shown in Fig. 3 a and b is provided. In the case shown in Fig. 3, the “bins” shown are actually the Voronoi partition cells corresponding to the data points involved in the 1-D cut.

(Q, ω) dispersion relation plots can be created by vertically stacking const-E 1-D cuts. Fig. 4 shows data for the (Q, ω) dispersion relation for magnons in MnF_2 from (0.5 0 0.05) to (1.5 0 0.05) (r.l.u.). Data from different final energy channels are normalized using $I \propto \Omega V_{\text{res}} = \Omega \frac{k_f^3}{\tan \theta_A}$ [19], where k_f is the wavevector of an energy channel, and θ_A is the half of the analyzer takeoff angle of the channel. It is worth mentioning though, that the normalization factor does not completely account for differences of the resolution ellipsoid of different final energy channels. Nevertheless, such (Q, ω) dispersion relation graph can be useful in visualizing dispersion relations on-the-fly in lieu of a full analysis accounting for resolution effects.

4. Conclusions and future work

While a lot of effort has been invested into the development and construction of MultiFLEXx and other multiplexing backends, software for data visualization lagged considerably behind. The extensive amount of variables commonly involved in MultiFLEXx experiments also makes data analysis prone to human error. This is especially true to on-the-fly analysis during experiments. Due to the widespread use of interpolation in graphics generation, users are tempted to perform scans with unnecessarily high data point density. The ambiguity of interpolated data can also lead to misinterpretations. All these factors can lead to sub-optimal utilization of beamtime. By providing the automated and unambiguous data aggregation and visualization toolbox *multiflexxlib*, the authors hope to address these difficulties and improve the scientific output of MultiFLEXx and other multiplexing backends that share a common concept [3,4,20]. The authors wish to add qualitative analysis functionalities such as modeling and fitting of experimental data in the future, which remains a challenging task [6].

Acknowledgments

This work is conducted under the collaboration between CIAE and HZB on the scientific usage of selected instruments at the research reactor BER II. This work receives financial support from China natural science fund grant no. 11227906. The authors express their thanks to everyone who contributed to the design, construction and commissioning of MultiFLEXx: Michael Rose, Bernd Urban, Werner Graf, Lutz Rossa, John Allibon, Felix Groitl, Diana Lucia Quintero-Castro, Zhilun Lu, Zita Hüsches, Manh Duc Le, Svyatoslav Alimov and Thomas Wilpert.

Conflict of interest

None.

Appendix. Usage example

```
import multiflexxlib as mfl # Imports the library
import matplotlib.pyplot as plt # Imports matplotlib, usually not
required if run from an interactive environment.
data = mfl.read_and_bin() # User will be prompted to choose a
folder containing data, as no path is provided in function argu-
ments.
data.summary() # Prints a list summarizing aggregated scan condi-
tions.
p = data.plot() # Plots all possible Const-E plots.
plt.show() # Shows the figure window. Usually not required if run
from an interactive environment.
```

References

- [1] Shen Y, Li Y-D, Wo H, Li Y, Shen S, Pan B, Wang Q, Walker HC, Steffens P, Boehm M, Hao Y, Quintero-Castro DL, Harriger LW, Frontzek MD, Hao L, Meng S, Zhang Q, Chen G, Zhao J. Nature 2016;540(7634):559. <http://dx.doi.org/10.1038/nature20614>.
- [2] Morris DJP, Tennant DA, Grigera SA, Klemke B, Castelnovo C, Moessner R, Czternasty C, Meissner M, Rule KC, Hoffmann J, Kiefer K, Gerischer S, Slobinsky D, Perry RS. Science 2009;326(5951):411–4. <http://dx.doi.org/10.1126/science.1178868>.
- [3] Freeman P, O. Birk J, Marko M, Bertelsen M, Larsen J, B. Christensen N, Lefmann K, Jacobsen J, Niedermayer C, Juranyi F, Rønnow H. In: EPJ Web Conferences, vol. 83, 2014, p. 03005. <http://dx.doi.org/10.1051/epjconf/20158303005>.
- [4] Groitl F, Graf D, Birk JO, Markó M, Bartkowiak M, Filges U, Niedermayer C, Rüegg C, Rønnow HM. Rev Sci Instrum 2016;87(3):035109. <http://dx.doi.org/10.1063/1.4943208>.
- [5] Le M, Quintero-Castro D, Toft-Petersen R, Groitl F, Skoulatos M, Rule K, Habicht K. Nucl Instrum Methods A 2013;729:220–6. <http://dx.doi.org/10.1016/j.nima.2013.07.007>.
- [6] Toft-Petersen R, Groitl F, Kure M, Lim J, Čermák P, Alimov S, Wilpert T, Le MD, Quintero-Castro D, Niedermayer C, Schneidewind A, Habicht K. Nucl Instrum Methods A 2016;830:338–44. <http://dx.doi.org/10.1016/j.nima.2016.06.004>.
- [7] Duc Le M, Skoulatos M, Quintero-Castro DL, Toft-Petersen R, Groitl F, Rule KC, Habicht K. Neutron News 2014;25(2):19–22. <http://dx.doi.org/10.1080/10448632.2014.902699>.
- [8] Arnold O, Bilheux J, Borreguero J, Buts A, Campbell S, Chapon L, Doucet M, Draper N, Leal RF, Gigg M, Lynch V, Markvardsen A, Mikkelsen D, Mikkelsen R, Miller R, Palmen K, Parker P, Passos G, Perring T, Peterson P, Ren S, Reuter M, Savici A, Taylor J, Taylor R, Tolchenov R, Zhou W, Zikovsky J. Nucl Instrum Methods A 2014;764:156–66. <http://dx.doi.org/10.1016/j.nima.2014.07.029>.
- [9] Ewings R, Buts A, Le M, van Duijn J, Bustinduy I, Perring T. Nucl Instrum Methods A 2016;834:132–42. <http://dx.doi.org/10.1016/j.nima.2016.07.036>.
- [10] Busing WR, Levy HA. Acta Crystallogr Sect A 1967;22(4):457–64. <http://dx.doi.org/10.1107/S0365110X67000970>.
- [11] McKinney W. In: van der Walt S, Millman J, editors. Proceedings of the 9th Python in Science Conference. 2010, p. 51–6.
- [12] Hunter JD. Comput Sci Eng 2007;9(3):90–5. <http://dx.doi.org/10.1109/MCSE.2007.55>.
- [13] Okazaki A, Turberfield K, Stevenson R. Phys Lett 1964;8(1):9–11. [http://dx.doi.org/10.1016/0031-9163\(64\)90774-7](http://dx.doi.org/10.1016/0031-9163(64)90774-7).
- [14] Bayrakci SP, Keller T, Habicht K, Keimer B. Science 2006;312(5782):1926–9. <http://dx.doi.org/10.1126/science.1127756>.
- [15] Barber CB, Dobkin DP, Huhdanpaa H. ACM Trans Math Software 1996;22(4):469–83. <http://dx.doi.org/10.1145/235815.235821>.
- [16] Paul S. plotmultiple manual, <https://github.com/nplot/nplot/blob/master/doc/pfcManual.pdf>. [Accessed: 12 April 2018].
- [17] Kempa M, Janousova B, Saroun J, Flores P, Boehm M, Demmel F, Kulda J. Physica B 2006;385–386:1080–2. <http://dx.doi.org/10.1016/j.physb.2006.05.371>.
- [18] Shirane G, Shapiro SM, Tranquada JM. Neutron scattering with a triple-axis spectrometer: basic techniques. Cambridge University Press; 2002.
- [19] Groitl F, Toft-Petersen R, Quintero-Castro DL, Meng S, Lu Z, Huesges Z, Le MD, Alimov S, Wilpert T, Kiefer K, Gerischer S, Bertin A, Habicht K. Sci Rep 2017;7(1):13637. <http://dx.doi.org/10.1038/s41598-017-14046-z>.
- [20] Lim JA, Siemensmeyer K, Čermák P, Lake B, Schneidewind A, Inosov DS. J Phys Conf Ser 2015;592(1):012145. <http://dx.doi.org/10.1088/1742-6596/592/1/012145>.