

Original software publication

# NeuTomPy toolbox, a Python package for tomographic data processing and reconstruction



Davide Micieli <sup>a,b,\*</sup>, Triestino Minniti <sup>c</sup>, Giuseppe Gorini <sup>b</sup>

<sup>a</sup> Dipartimento di Fisica, Università della Calabria, via Pietro Bucci, 87036 Arcavacata di Rende (Cosenza), Italy

<sup>b</sup> Dipartimento di Fisica 'G. Occhialini', Università degli Studi Milano-Bicocca, Piazza della Scienza 3, 20126 Milano, Italy

<sup>c</sup> STFC, Rutherford Appleton Laboratory, ISIS Facility, Harwell, OX11 0QX, United Kingdom

## ARTICLE INFO

### Article history:

Received 21 September 2018

Received in revised form 12 November 2018

Accepted 7 January 2019

### Keywords:

Tomographic reconstruction software

Tomography

Neutron imaging

## ABSTRACT

In this article we present the NeuTomPy Toolbox, a new Python package for tomographic data processing and reconstruction. The toolbox includes pre-processing algorithms, artifacts removal and a wide range of iterative reconstruction methods as well as the Filtered Back Projection algorithm. The NeuTomPy toolbox was conceived primarily for neutron tomography datasets and developed to support the need of users and researchers to compare state-of-the-art reconstruction methods and choose the optimal data processing workflow for their data. In fact, in several cases sparse-view datasets are acquired to reduce scan time during a neutron tomography experiment. Hence, there is great interest in improving quality of the reconstructed images by means of iterative methods and advanced image-processing algorithms. The toolbox has a modular design, multi-threading capabilities and it supports Windows, Linux and Mac OS operating systems. The NeuTomPy toolbox is open source and it is released under the GNU General Public License v3, encouraging researchers and developers to contribute. In this paper we present an overview of the main toolbox functionalities and finally we show a typical usage example.

© 2019 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	v1.0.8
Permanent link to code/repository used for this code version	<a href="https://github.com/ElsevierSoftwareX/SOFTX_2018_179">https://github.com/ElsevierSoftwareX/SOFTX_2018_179</a>
Legal Code License	GNU General Public License v3.0
Code versioning system used	Git
Software code languages, tools, and services used	Python 3.5 or 3.6
Compilation requirements, operating environments & dependencies	Linux, Windows and Mac OS
If available Link to developer documentation/manual	<a href="https://neutomp-toolbox.readthedocs.io">https://neutomp-toolbox.readthedocs.io</a>
Support email for questions	<a href="mailto:davide.micieli@unical.it">davide.micieli@unical.it</a>

## 1. Motivation and significance

Neutron Tomography (NT) has become a routine method at many neutron sources to non-destructively investigate the inner structure of a wide range of objects. The commercial software Octopus [1] by Inside Matters is a well established tool for reconstruction of tomographic data at neutron imaging beamlines. However, this software requires a significant investment and generally users

can perform a preliminary data processing with Octopus only at the imaging facility. Data analysis is a crucial step for the output of an experiment, so users usually spend time to optimize the data processing mainly at home. This poses a strong demand of freeware and powerful tools to perform data processing of neutron data.

Image acquisition in NT is very time-consuming with respect to X-ray Computed Tomography (CT) and, in several cases, under-sampled datasets are acquired to reduce the scan time and optimize beamtime usage during an experiment. The widely used Filtered Back Projection (FBP) algorithm generates reconstructed images affected by aliasing artifacts when the number of projections does not satisfy the Nyquist–Shannon condition [2]. Iterative reconstruction methods generally outperform analytical methods,

\* Corresponding author at: Dipartimento di Fisica, Università della Calabria, via Pietro Bucci, 87036 Arcavacata di Rende (Cosenza), Italy.

E-mail addresses: [davide.micieli@unical.it](mailto:davide.micieli@unical.it) (D. Micieli), [triestino.minniti@stfc.ac.uk](mailto:triestino.minniti@stfc.ac.uk) (T. Minniti).

such as FBP, to handle under-sampled datasets [3]. Octopus software provides only two reconstruction methods: the FBP and the Simultaneous Algebraic Reconstruction Technique (SART). Modern reconstruction methods are not implemented. On the other hand, several open source tools for tomographic reconstruction are available nowadays but they are mainly developed for X-ray CT and they are not ready to handle neutron data. Some image pre-processing algorithms are mandatory in NT to obtain accurate reconstruction, i.e. the estimation of the rotation axis tilt and the related registration of the projections, the suppression of gamma-spots and the data normalization with respect to the radiation dose. Reconstruction tools for X-ray CT generally include some, but not all of such correction algorithms. For example, the ASTRA toolbox [4] is a Matlab and Python package that provides highly efficient implementation of iterative methods for CPUs and GPUs. ASTRA toolbox is only focused on the reconstruction step and it does not include any pre-processing, post-processing algorithms or functions to read and write data. On the other hand, the Python package TomoPy [5] includes several pre-processing and post-processing algorithms and provides implementation for CPUs of a wide range of iterative reconstruction methods. Moreover TomoPy is not ready to handle neutron data, since it does not include functions to estimate the rotation axis tilt and to compute the related correction on projection data. Furthermore, TomoPy is available only for Linux and Mac OS operating systems. MuhRec [6] is the only free software that was conceived for NT. It includes several filters and pre-processing algorithms and it is currently the main free alternative to Octopus for data processing of neutron data. However, at time of writing, MuhRec does not provide any iterative reconstruction method support.

In this paper we present the NeuTomPy Toolbox, a new Python package for tomographic data processing, that is specifically designed to compensate the shortcomings of the aforementioned software tools. The NeuTomPy toolbox was conceived primarily for NT and developed to support the need of users and researchers to compare state-of-the-art reconstruction methods and choose the optimal data processing workflow for their data. The toolbox has a modular design, multi-threading capabilities and it supports Windows, Linux and Mac OS operating systems. The NeuTomPy toolbox is open source and it is released under the GNU General Public License v3, allowing users to freely use it and encouraging researchers and developers to contribute. Previously, this package has been used for comparative studies [3,7] of reconstruction methods in NT and now is freely distributed to the neutron imaging community.

## 2. Software description

Here we describe the architecture of NeuTomPy Toolbox and present its main functionalities.

### 2.1. Software architecture

The NeuTomPy toolbox is written in Python. We chose this programming language because it is open-source, cross-platform, human-readable and allows researchers to use and contribute to it easily. The toolbox is divided into several sub-modules, each of these represents a particular phase of a typical CT reconstruction pipeline. The entire chain is represented in Fig. 1. The NeuTomPy toolbox exploits several Python libraries for scientific computing and image processing, i.e. NumPy [8], NumExpr [9], SciPy [10], scikit-image [11], OpenCV [12] and SimpleITK [13]. In particular, the CT reconstruction step is powered by the ASTRA Toolbox. NeuTomPy combined with ITK-SNAP [14] or 3D Slicer [15] turns out to be a complete open-source software suite for CT.

### 2.2. Software functionalities and sample code snippets

The NeuTomPy toolbox allows to perform the steps of a typical CT reconstruction workflow (Fig. 1). The first task is represented by the reading of a raw dataset. The implemented reader handles TIFF and FITS files and converts a stack of images into a numpy array. A dataset containing raw projections, dark-field, flat-field images and the projection at 180° can be read by:

```
import neutompy as ntp
proj, dark, flat, proj_180 = ntp.read_dataset(proj_180=
True)
```

hence the user can select the data to read from a dialog box. Subsequently, the projection data must be normalized with respect to dark-field and flat-field images to compute the transmission images. If the source intensity is not stable the images can be normalized with respect to the radiation dose [3]. In this case, the user must specify a region of interest (ROI) which corresponds to a background area not covered by the specimen in all the projections (we called it the dose ROI). It can be specified in three different ways: drawing interactively a rectangular selection, specifying the ROI' s coordinates or reading an ImageJ .roi file. For example, to normalize data and select interactively the dose ROI, the Python instruction is:

```
norm, norm_180 = ntp.normalize_proj(proj, dark, flat,
proj_180=proj_180,
dose_draw=True)
```

where the function `normalize_proj` returns a 3D array containing the stack of normalized projections (`norm`) and a 2D array representing the normalized radiograph at 180° (`norm_180`).

A common experimental issue in NT is the misalignment of the rotation axis with respect to the vertical axis of the detector. The function `correction_COR` evaluates the horizontal offset and the tilt angle by minimizing the squared error between two opposite radiographs computed at different vertical positions, as described in [6], and finally it registers all the projections. The Python instruction for this task is:

```
norm = ntp.correction_COR(norm, proj_0, proj_180)
```

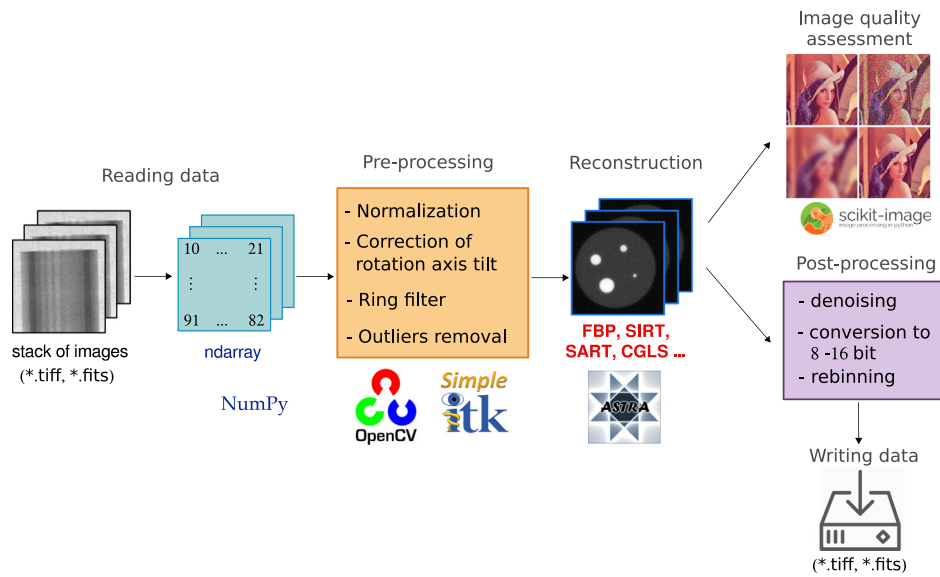
where `proj_0` and `proj_180` are the projections (raw or normalized) at 0° and 180°, respectively. The user selects interactively different ROIs where the sample is visible. Subsequently the results and some information about the evaluation of the rotation axis correction are shown. We report in Fig. 2 an example for the rotation axis correction: the difference of the projections at 0° ( $P_0$ ) and the mirrored projection at 180° ( $P_{\pi}^{flipped}$ ) before and after the correction are shown in the left and right side, respectively.

The NeuTomPy toolbox includes an outlier removal which replaces a pixel value by the median of the neighborhood pixels if it deviates from the median by more than a certain value. This threshold value can be specified by the user as a global value or proportional to the local standard deviation. It is provided also a destriping filter, based on combined wavelet and Fourier analysis, to suppress the ring artifacts [16].

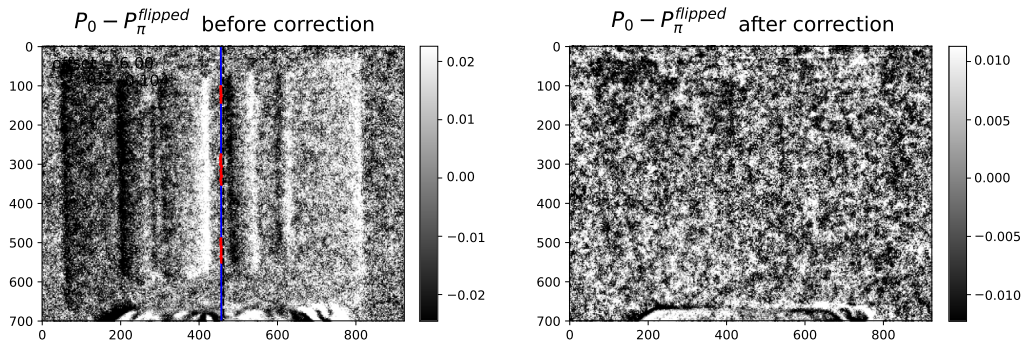
The reconstruction module includes all CPU- and GPU-based algorithms for 2D parallel beam geometry implemented in the ASTRA toolbox and some additional reconstruction methods distributed as ASTRA plugins. The available algorithms are summarized in Table 1. The instruction to perform a CT reconstruction is the following:

```
rec = ntp.reconstruct(norm, angles, method, parameters)
```

where `rec` is the reconstructed volume, `angles` is one-dimensional array containing the view angles in radians, `method` is a string which indicates the algorithm to use and `parameters` is a Python dictionary that contains specific settings of the



**Fig. 1.** Diagram representing the typical CT data processing steps that can be performed by NeuTomPy toolbox. The package has a modular structure that follows the data processing chain.



**Fig. 2.** Results of the rotation axis correction: the difference  $P_0 - P_\pi^{\text{flipped}}$  before (left) and after (right) the correction. The rotation axis is determined correctly if the difference image  $P_0 - P_\pi^{\text{flipped}}$  after correction does not contain sample features.

reconstruction algorithm. The allowed values for method and parameters follow the convention of the ASTRA toolbox, reported in the documentation [17]. For example, the following instruction is used to compute with GPU support a FBP reconstruction with the Hamming filter:

```
rec = ntp.reconstruct(norm, angles, method="FBP_CUDA",
                    parameters={"FilterType": "hamming"})
```

while a SIRT reconstruction with 100 iterations and pixel values limited in the range [0, 2] can be performed by:

```
rec = ntp.reconstruct(norm, angles, method="SIRT_CUDA",
                    parameters={"iterations": 100, "MinConstraint": 0.0,
                               "MaxConstraint": 2.0}).
```

The NeuTomPy toolbox allows to compare and evaluate the performance of different reconstruction algorithms in terms of several image quality indexes. The metrics implemented are the Contrast-to-Noise-Ratio (CNR) [3], the Normalized Root Mean Square Error (NRMSE) [3], an edge quality metric [3] and the Structural Similarity Index (SSIM) [18].

### 3. Illustrative examples

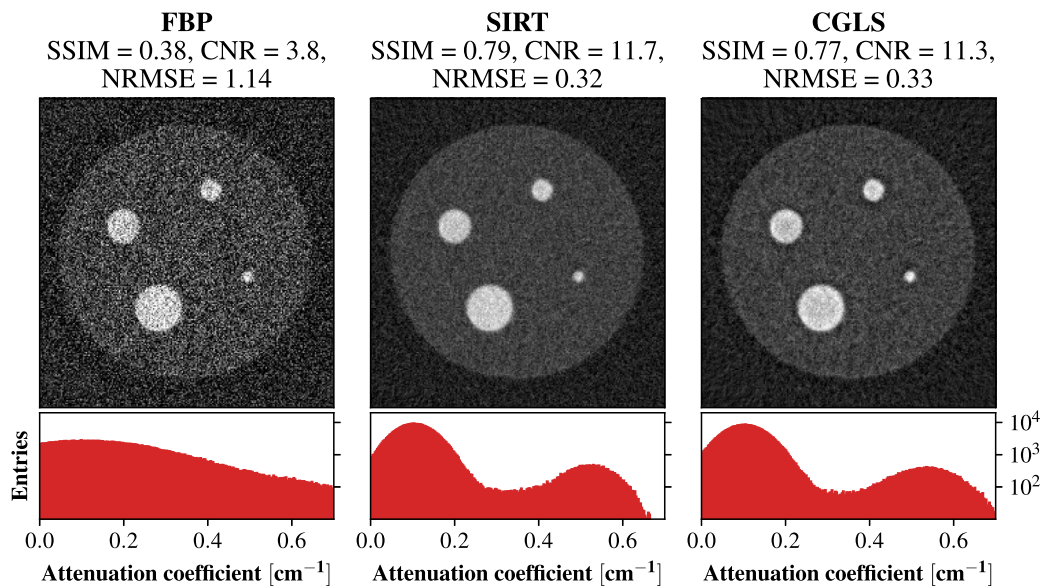
Here we demonstrate the possibility to perform several reconstruction algorithms and compare them quantitatively using the NeuTomPy toolbox. We used neutron images of a phantom sample

**Table 1**

List of the CT reconstruction methods included in NeuTomPy Toolbox for two-dimensional parallel-beam geometries.

Method	CPU	GPU
BP [2]	x	x
FBP [2]	x	x
ART [2]	x	
SART [2]	x	x
CGLS [19]	x	x
SIRT [20]	x	x
NN-FBP [21]	x	x
MR-FBP [22]	x	x

acquired at the IMAT beamline [23,24], ISIS neutron spallation source, UK. The phantom, already analyzed in a previous work [3], is an aluminium cylinder containing four holes of different diameters and filled with iron powder. We used for CT reconstruction an under-sampled dataset with 1/3 of the number of projections required by the Nyquist–Shannon condition. We performed FBP, SIRT and CGLS reconstructions and we compare them in terms of the image quality indexes NRMSE, SSIM and CNR. We consider the SIRT reconstruction (200 iterations) of a full-view dataset, which is sampled to fulfill the Nyquist–Shannon condition, as the reference image for the computation of the NRMSE and SSIM. The CNR was computed considering a ROI that includes one iron rod and with the



**Fig. 3.** A comparison of reconstructed images of a phantom sample, obtained using FBP, SIRT (200 iterations) and CGLS (10 iterations). Below each image the histogram of the attenuation coefficient values within the sample is represented in the range  $[0, 0.7] \text{ cm}^{-1}$ .

second ROI outside the sample. The results are shown in Fig. 3. It is clear that the two iterative algorithms outperform the FBP method. In fact, the CGLS and the SIRT reconstructions have higher CNR and SSIM, and lower NRMSE than the FBP, which indicate better image quality. In general, the under-sampling and the noise in the projection data cause in the reconstructed images a broadening of the attenuation coefficients distribution. However, unlike FBP reconstruction, the CGLS and the SIRT images are characterized by a bimodal distribution of the gray values, which reflects the composition of the sample.

The source code of this analysis is omitted here for brevity. However, the source code for this and other examples can be found in the GitHub repository.

#### 4. Impact

Data processing is the last step of a NT experiment but it is crucial for the interpretation of the results. Advanced image processing algorithms can extract hidden information from data and reduce the tomographic scan time. Hence new software tools, specifically designed for neutron data, are required to compare state-of-the-art image processing algorithms. Working on robust methods and tools to improve image quality means get better output from NT experiments. However, state-of-the-art iterative reconstruction methods are not implemented in Octopus and MuhRec, which are the leading software for NT reconstruction. The NeuTomPy toolbox solves this shortcoming because it is ready to work with neutron data and allows to perform and compare several iterative reconstruction methods. Researches can define the optimal data processing workflow for their specific problem using the NeuTomPy toolbox. The code is open-source, hence developers and researchers are invited to contribute.

#### 5. Conclusions

In this paper we presented the NeuTomPy Toolbox, a new Python package for tomographic data processing. We demonstrated that the toolbox is ready to work with neutron data and allows researchers to state the optimal data processing workflow for their specific investigation. The first release includes pre-processing algorithms, artifacts removal and a wide range of classical and state-of-the-art reconstruction methods. The NeuTomPy

toolbox supports Windows, Linux and Mac OS operating systems and it is released as open source. Researchers can freely use it and contribute to the project.

The future development will involve improvement of pre-processing algorithms (e.g. scattering correction), addition of new reconstruction methods and finally the implementation of a Graphical User Interface (GUI).

#### Acknowledgments

We acknowledge the CNR-STFC Agreement 2014–2020 (N. 3420 2014–2020) concerning collaboration in scientific research at the ISIS Spallation Neutron Source, for financial support.

#### References

- [1] Inside Matters. Octopus imaging software, <https://octopusimaging.eu/>.
- [2] Kak AC, Slaney M. *Principles of computerized tomographic imaging*. Society for Industrial and Applied Mathematics; 2001.
- [3] Micieli D, Minniti T, Formoso V, Kockelmann W, Gorini G. A comparative study of reconstruction methods applied to neutron tomography. *J Instrum* 2018;13(06):C06006. <http://dx.doi.org/10.1088/1748-0221/13/06/C06006>.
- [4] van Aarle W, Palenstijn WJ, Cant J, Janssens E, Bleichrodt F, Dabrovolski A, Beenhouwer JD, Batenburg KJ, Sijbers J. Fast and flexible X-ray tomography using the ASTRA toolbox. *Opt. Express* 2016;24(22):25129–47. <http://dx.doi.org/10.1364/OE.24.025129>.
- [5] Gürsoy D, De Carlo F, Xiao X, Jacobsen C. Tomopy: a framework for the analysis of synchrotron tomographic data. *J Synchrotron Radiat* 2014;21(5):1188–93. <http://dx.doi.org/10.1107/S1600577514013939>.
- [6] Kaestner AP. MuhRec—new tomography reconstructor. *Nucl Instrum Methods Phys Res A* 2011;651(1):156–60. <http://dx.doi.org/10.1016/j.nima.2011.01.129>.
- [7] Micieli D, Minniti T, Evans LM, Gorini G. Accelerating neutron tomography experiments through artificial neural network based reconstruction. *Sci. Rep.* 2019;9(2450). <http://dx.doi.org/10.1038/s41598-019-38903-1>.
- [8] van der Walt S, Colbert SC, Varoquaux G. The numpy array: a structure for efficient numerical computation. *Comput Sci Eng* 2011;13(2):22–30. <http://dx.doi.org/10.1109/MCSE.2011.37>.
- [9] NumExpr documentation. 2018, <https://numexpr.readthedocs.io/>.
- [10] Jones E, Oliphant T, Peterson P, et al. *SciPy*. 2001.
- [11] van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu Ta. scikit-image: image processing in Python. *PeerJ* 2014;2:e453. <http://dx.doi.org/10.7717/peerj.453>.
- [12] Bradski G. *The opencv library*. Dr. Dobb's J. *Softw. Tools* 2000.
- [13] Yaniv Z, Lowekamp BC, Johnson HJ, Beare R. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *J. Digit. Imaging* 2018;31(3):290–303. <http://dx.doi.org/10.1007/s10278-017-0037-8>.

- [14] Yushkevich PA, Piven J, Cody Hazlett H, Gimpel Smith R, Ho S, Gee JC, Gerig G. User-Guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage* 2006;31(3):1116–28.
- [15] Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin J-C, Pujol S, Bauer C, Jennings D, Fennessy F, Sonka M, Buatti J, Aylward S, Miller JV, Pieper S, Kikinis R. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn. Reson. Imaging* 2012;30(9):1323–41. <http://dx.doi.org/10.1016/j.mri.2012.05.001>.
- [16] Münch B, Trtik P, Marone F, Stampanoni M. Stripe and ring artifact removal with combined wavelet-fourier filtering. *Opt Express* 2009;17(10):8567. <http://dx.doi.org/10.1364/OE.17.008567>.
- [17] ASTRA toolbox documentation. 2018, <https://www.astra-toolbox.com/docs/algs/index.html>.
- [18] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 2004;13(4):600–12. <http://dx.doi.org/10.1109/TIP.2003.819861>.
- [19] Scales Ja. Tomographic inversion via the conjugate gradient method. *Geophysics* 1987;52(2):179. <http://dx.doi.org/10.1190/1.1442293>.
- [20] Gilbert P. Iterative methods for the three-dimensional reconstruction of an object from projections. *J Theoret Biol* 1972;36(1):105–17. [http://dx.doi.org/10.1016/0022-5193\(72\)90180-4](http://dx.doi.org/10.1016/0022-5193(72)90180-4).
- [21] Pelt DM, Batenburg KJ. Fast tomographic reconstruction from limited data using artificial neural networks. *IEEE Trans Image Process* 2013;22(12):5238–51. <http://dx.doi.org/10.1109/TIP.2013.2283142>.
- [22] Pelt DM, Batenburg KJ. Improving filtered backprojection reconstruction by data-dependent filtering. *IEEE Trans Image Process* 2014;23(11):4750–62. <http://dx.doi.org/10.1109/TIP.2014.2341971>.
- [23] Minniti T, Watanabe K, Burca G, Pooley DE, Kockelmann W. Characterization of the new neutron imaging and materials science facility imat. *Nucl Instrum Methods Phys Res A* 2018;888:184–95. <http://dx.doi.org/10.1016/j.nima.2018.01.037>.
- [24] Minniti T, et al. Materials analysis opportunities on the new neutron imaging facility IMAT@ISIS. *J. Instr.* 2016;11:C03014. <http://dx.doi.org/10.1088/1748-0221/11/03/C03014>.