



## Original software publication

## MJOLNIR: A software package for multiplexing neutron spectrometers

Jakob Lass<sup>a,b,\*</sup>, Henrik Jacobsen<sup>a</sup>, Daniel G. Mazzone<sup>a</sup>, Kim Lefmann<sup>b</sup><sup>a</sup> Laboratory for Neutron Scattering and Imaging, Paul Scherrer Institut, CH-5232, Villigen, Switzerland<sup>b</sup> Nanoscience Center, Niels Bohr Institute, University of Copenhagen, DK-2100, Copenhagen Ø, Denmark

## ARTICLE INFO

## Article history:

Received 9 July 2020

Received in revised form 24 September 2020

Accepted 24 September 2020

## Keywords:

Inelastic neutron scattering

Three-axis spectroscopy

Visualisation tool

## ABSTRACT

Novel multiplexing triple-axis neutron scattering spectrometers yield significant improvements to the common triple-axis concept. While the planar scattering geometry keeps ensuring compatibility with complex sample environments, a simultaneous detection of scattered neutrons at various angles and energies leads to tremendous improvements in the data acquisition rate. Here we report on the software package MJOLNIR that we have developed to handle the resulting enhancement in data complexity. Using data from the new CAMEA spectrometer of the Swiss Spallation Neutron Source at the Paul Scherrer Institut, we show how the software reduces, visualises and treats observables measured on a multiplexing spectrometer. The software package has been generalised to a uniform framework, allowing for collaborations across multiplexing instruments at different facilities, further facilitating new developments in data treatment, such as fitting routines and modelling of multi-dimensional data.

© 2020 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	1.1.13	0.8.1
Permanent link to code/repository used for this code version	<a href="https://github.com/ElsevierSoftwareX/SOFTX-D-20-00002">https://github.com/ElsevierSoftwareX/SOFTX-D-20-00002</a>	<a href="https://github.com/ElsevierSoftwareX/SOFTX-D-20-00002">https://github.com/ElsevierSoftwareX/SOFTX-D-20-00002</a>
Code Ocean compute capsule	N/A	N/A
Legal Code License	Mozilla Public License 2.0 (MPL-2.0)	GPL-3.0 License
Code versioning system used	git	git
Software code languages, tools, and services used	Python, scipy, matplotlib, ufit	Python, MJOLNIR, PyQt5, fbs
Compilation requirements, operating environments & dependencies		MJOLNIR
If available Link to developer documentation/manual	<a href="https://mjolnir.readthedocs.io/">https://mjolnir.readthedocs.io/</a>	<a href="https://www.psi.ch/en/sinq/comea/data-treatment">https://www.psi.ch/en/sinq/comea/data-treatment</a>
Support email for questions	<a href="mailto:mjolnirpackage@gmail.com">mjolnirpackage@gmail.com</a>	<a href="mailto:mjolnirpackage@gmail.com">mjolnirpackage@gmail.com</a>

## Software metadata

Current software version	1.1.13	0.8.1
Permanent link to executables of this version	N/A	<a href="https://www.psi.ch/en/sinq/comea/data-treatment">https://www.psi.ch/en/sinq/comea/data-treatment</a>
Legal Software License	Mozilla Public License 2.0 (MPL-2.0)	GPL-3.0 License
Computing platforms/Operating Systems	Linux, OS X, Microsoft Windows, Unix-like	Linux, OS X, Microsoft Windows, Unix-like
Installation requirements & dependencies	Python	Python & PyQt5
If available, link to user manual – if formally published	<a href="https://mjolnir.readthedocs.io/">https://mjolnir.readthedocs.io/</a>	<a href="https://www.psi.ch/en/sinq/comea/data-treatment">https://www.psi.ch/en/sinq/comea/data-treatment</a>
include a reference to the publication in the reference list		
Support email for questions	<a href="mailto:mjolnirpackage@gmail.com">mjolnirpackage@gmail.com</a>	<a href="mailto:mjolnirpackage@gmail.com">mjolnirpackage@gmail.com</a>

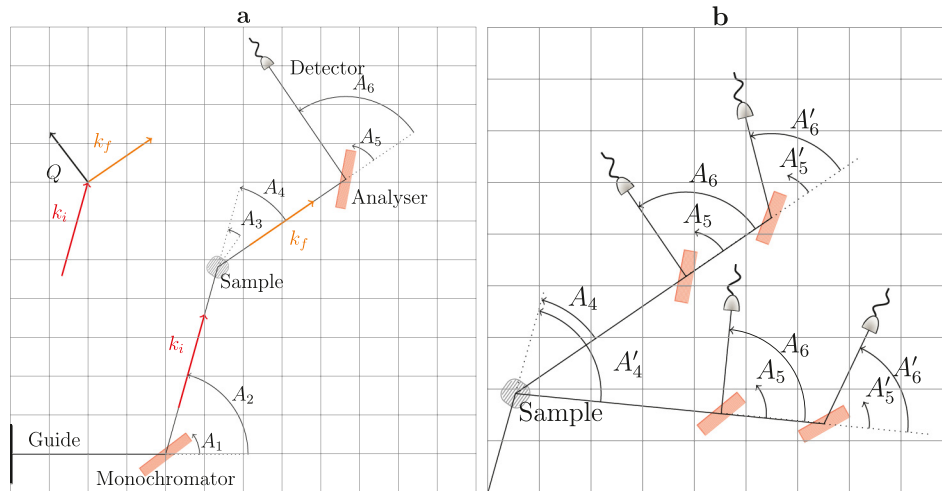
## 1. Introduction

Inelastic neutron scattering instruments allow detailed studies of the dynamical structure factor,  $S(\mathbf{Q}, \omega)$ , where  $\mathbf{Q}$  is a scattering

vector in reciprocal space and  $\hbar\omega = \Delta E$  an energy transfer. One of the work horses of modern neutron scattering is the triple-axis instrument (see Fig. 1), which typically have a high neutron flux and good energy resolution [1].

Standard triple-axis (TAS) instruments cover only a single  $(\mathbf{Q}, \hbar\omega)$ -position per acquisition time, which leads to isolated trajectories mapped along lines of  $\omega$  or  $\mathbf{Q}$  during experiments.

\* Correspondence to: 5232 Villigen PSI, Switzerland.  
E-mail address: [jakob.lass@nbi.ku.dk](mailto:jakob.lass@nbi.ku.dk) (J. Lass).



**Fig. 1.** a: Sketch of a standard triple-axis instrument depicting the six angles,  $A_1$ – $A_6$ . b: Zoom on multiplexing secondary spectrometer with two scattering angles,  $A_4$  and  $A_4'$ , and two monochromators at different rotation angles,  $A_5$  and  $A_6$ , and  $A_5'$  and  $A_6'$ , respectively, giving two simultaneously measured final energies.

Multiplexing triple-axis instruments extend the triple-axis concept by employing multiple analysers and detectors. This allows for simultaneous measurements of large  $S(\mathbf{Q}, \omega)$  regions, while preserving a high neutron flux. This is in contrast to direct geometry time-of-flight mapping instruments, which have an even larger coverage of reciprocal space, but pay the price in form of a significant reduction in incoming flux.

The latest addition to the growing list of multiplexing triple-axis spectrometers around the world is the new secondary neutron spectrometer CAMEA@PSI [2] at the Paul Scherrer Institut (PSI), Switzerland. Others are RITA-II [3], FlatCONE [4], UFO [5], MultiFLEXX [6], PUMA [7], Bambus [8], and MACS [9]. In some sense the usage of most of these instruments is simplified through the reduction of movable parts and parameters, while post processing of data has increased significantly in complexity. This increased complexity is also found when using direct-geometry time-of-flight spectrometers, but is a comparatively new challenge for TAS. The point clouds measured on these spectrometers have to be treated in a fundamentally new fashion, therefore pushing the data reduction onto a new complexity level.

For some of these multiplexing instruments, specialised data treatment software have been written already. This includes MultiFLEXXlib [10] for MultiFLEXX and nplot for FlatCone [11]. Unfortunately, the design of CAMEA@PSI is more complex, falling outside of the scope of these packages. This is either because the software assumes a perfect instrument alignment using geometrical calculations of scattering angles and final energies, or because multiple signals on the same detector cannot be treated.

Other popular software packages include Horace [12] (written in Matlab) which is used on direct time-of-flight neutron scattering spectrometers; Dave [13] (written in IDL), which can handle data from multiple different neutron scattering instruments, including standard triple-axis spectrometers and time-of-flight spectrometers; and MANTID [14] (written in C++ and Python) that was specifically developed for time-of-flight spectrometers. However, neither of these programs have been written to handle data acquired on multiplexing triple-axis spectrometers.

Here, we present the software package MJOLNIR, which has been primarily developed for CAMEA@PSI. However, the software can handle data from every multiplexing triple-axis neutron spectrometer, enabling utility across facilities. MJOLNIR offers tools to quickly convert data from detector counts in the instrumental geometry to reciprocal space, and to visualise the data in 1, 2 and 3 dimensions. The software has been written with focus on

being user-friendly, offering a scripting, a command line, and a graphical interface.

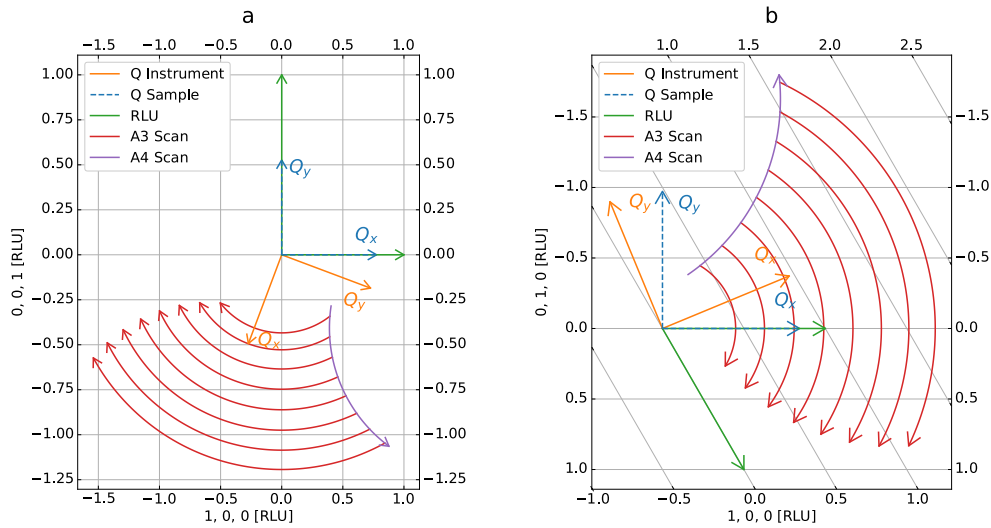
MJOLNIR has been coded in the open source language Python and can be downloaded at the PSI website [15]. MJOLNIR makes use of fundamental packages for scientific computing in Python, such as matplotlib [16], scipy [17], and its sub-package numpy, and Pandas [18]. As such the software is radially distributable and compatible across platforms through Python's package interface (PyPI). We mention that a trend for academic software towards Python is currently recognisable. In fact, well-established Matlab [19] programs such as SpinW [20] are being rewritten to allow for Python bindings. New large scale facilities including the European Spallation Source [21] plan to run with Python-based systems [14].

In this article we present the main features of MJOLNIR. The full documentation including extensive tutorials is available at the PSI website [15]. Before presenting MJOLNIR we first explain the standard triple-axis spectrometer concept and its differences to multiplexing instruments. We then describe the different coordinate systems used for in the data treatment. Subsequently, we describe the structure of MJOLNIR, followed by examples of some of its features.

### 1.1. Triple axis spectrometers

Fig. 1a shows a sketch of a typical triple-axis spectrometer with the six relevant rotation axes,  $A_1$ – $A_6$ . The monochromator is rotated with respect to the direct beam by  $A_1$ . Through the Bragg scattering condition the sample is positioned at  $A_2 = 2A_1$ , such that a single incoming neutron wavevector,  $\mathbf{k}_i$  (and thereby the energy  $E_i$ ) is selected. The sample rotation is denoted  $A_3$ , while  $A_4$  is the angle between the incident beam on the sample and the analyser. Similarly to the monochromator, the analyser is rotated by  $A_5$  and the detector is placed at  $A_6 = 2A_5$  to select a single final neutron wavevector,  $\mathbf{k}_f$  (and thereby the energy  $E_f$ ). The momentum and energy transfer between the neutron and the sample are given by  $\mathbf{Q} = \mathbf{k}_i - \mathbf{k}_f$  and  $\Delta E = E_i - E_f$ , which can be calculated from  $A_1 - A_6$  and the crystallographic properties of the monochromator and analyser. We refer to Ref. [1] for a more extensive description of triple-axis spectrometers.

A multiplexing triple-axis instrument uses multiple analysers and detectors which leads to simultaneous measurements of the neutron scattering intensity at multiple  $A_4 - A_6$  positions (see Fig. 1b). On most multiplexing instruments the  $A_5$  and  $A_6$  angles are fixed. At some instruments position sensitive detectors



**Fig. 2.** Overview of coordinate axes used for projection and RLU axes for **a:** tetragonal  $\text{MnF}_2$  and **b:** hexagonal  $\text{YMnO}_3$ . The  $A_3$  scan lines cover  $90^\circ$  and have constant  $A_4$  values between  $-16^\circ$  and  $-72^\circ$  covered by a single instrument setting by CAMEA@PSI.

collect neutrons from multiple analysers. In addition, on most multiplexing instruments (including CAMEA@PSI) the scattering direction off the analysers is rotated by  $90^\circ$  when compared to standard triple-axis instruments, thus scattering neutrons onto detectors that are out-of-plane. This change is employed for practical reasons, to allow for a maximal density of analyser-detector pairs. This rotation affects the resolution ellipsoid, rendering the current analytical schemes inapplicable [22]. Appropriate calculations of the resolution function in multiplexing geometries are under development. In general, the energy resolution of multiplexing spectrometers is comparable to the ones on standard triple-axis instruments. At CAMEA@PSI a resolution of between 80 and 280  $\mu\text{eV}$  at Full Width at Half Maximum (FWHM) has been reported [2].

### 1.2. Coordinate systems

In MJOLNIR, the data treatment is facilitated by defining several related coordinate systems that describe the scattering vector,  $\mathbf{Q}$ , illustrated in Fig. 2.

The instrument coordinate system is an orthonormal coordinate system that is sample independent. It is defined by  $\mathbf{C}_{\text{instr}} = (\mathbf{Q}_x, \mathbf{Q}_y, \mathbf{Q}_z, \Delta E)$ , where  $\mathbf{Q}_y$  is parallel to  $\mathbf{k}_i$  and orthogonal to  $\mathbf{Q}_x$  within the scattering plane. The out-of-plane component  $\mathbf{Q}_z$  is zero in all cases.

The sample coordinate system shares the axes of the instrument coordinate system, but is rotated so that the first axis points along a reciprocal lattice vector of the sample:  $\mathbf{C}_{\text{sample}} = (\tilde{\mathbf{Q}}_x, \tilde{\mathbf{Q}}_y, \tilde{\mathbf{Q}}_z, \Delta E)$ .

The reciprocal lattice coordinate system is defined by two reciprocal lattice vectors in the scattering plane,  $\mathbf{C}_{\text{RLU}} = (\mathbf{P}_1, \mathbf{P}_2, \Delta E)$ , which is often chosen to be along a high symmetry direction of the sample. For samples with cubic, tetragonal, or orthorhombic symmetry, the reciprocal lattice coordinate system overlaps with the sample coordinate system, such as illustrated in Fig. 2a. In contrast, the reciprocal lattice vectors are not orthogonal in hexagonal, trigonal, monoclinic or triclinic systems, see Fig. 2b.

A typical measurement on CAMEA@PSI consists of rotating  $A_3$  while keeping all other angles constant. This generates data points in the RLU coordinate system as shown in Fig. 2 by the red lines.

## 2. The scope of MJOLNIR

MJOLNIR is designed to provide tools for three purposes: (1) conversion of data into the reciprocal lattice coordinate system and the sample coordinate system. (2) Normalisation and visualisation of the data in 1, 2 and 3 dimensions, and (3) tools to aid the analysis of the data. We now describe the scope of MJOLNIR in these three aspects.

### 2.1. Data conversion

Conversion of data to the reciprocal lattice coordinate system requires detailed knowledge of the specific instrument setup. MJOLNIR has initially been developed for CAMEA@PSI, and thus includes an accurate description of this instrument. We aim for MJOLNIR to be used across many multiplexing spectrometers including MultiFLEX [23], FlatCone [4] and Bambus [8].

There are two methods by which the required instrument parameters can be extracted for the data conversion, which are explained in detail in Ref. [2]. The parameters can either be calculated from the instrument design directly, assuming a perfect setup, or measured through an experiment allowing for small deviations. A direct measurement yields more accurate results, but requires some extra modelling. At CAMEA@PSI, the scattering angles are calculated, while final energies are measured directly. At MultiFLEX both parameters are calculated [23]. MJOLNIR supports both options.

Further differences among various multiplexing instruments arise from the fact that some of them employ a prismatic analyser concept where several energies are scattered from one analyser [24], while others do not. However, by implementing the prismatic concept, the standard setup will be a limiting case of just a single energy, and is thus also supported.

The requirements for the upcoming indirect time-of-flight spectrometer BIFROST at the European Spallation Source are worth noting [25,26]. The instrument shares many similarities with CAMEA@PSI; both possess multiplexing secondary spectrometers, utilising the prismatic concept in combination with position sensitive detectors. However, their primary spectrometers differ. CAMEA@PSI is situated at a quasi-continuous source using a monochromator to determine the incoming energy. BIFROST, on the other hand, will be situated on a pulsed source, determining the incoming energy by time-of-flight. In effect,

BIFROST performs a scan over incoming energies within a defined window, whereas CAMEA@PSI employs one fixed incoming energy only. In practice, this extends BIFROST data by an additional dimension when compared to CAMEA@PSI. We expect that extensions of MJOLNIR will be suitable for handling data from BIFROST.

## 2.2. Visualisation

MJOLNIR is designed to work directly with raw data, where each data point consists of a neutron count, a monitor count, and normalisation value, as well as a position in reciprocal space. Acquiring multiple scans at almost identical positions will increase the likelihood that multiple points are in close vicinity of each other. One may argue that if their difference is smaller than the expected instrumental uncertainty they should be binned. The approach, however, imposes a resolution estimate reducing the transparency of the data process, as two points within a given distance in a certain parameter space might be binned in some part of the measurement volume but not in other parts. This is due to the fact that the instrument resolution changes across the probed volume of reciprocal space. Instead, we prefer to treat all data points separate as long as no user-defined tolerance is provided.

MJOLNIR provides tools for visualising data from any multiplexing instrument in 1, 2 and 3 dimensions in the reciprocal lattice coordinate system and in the sample coordinate system. For these purposes, the user defines the size of the bins as well as the directions of the desired cuts. MJOLNIR then bins and normalises the raw data to the monitor count, and plots the binned data. The tools are described in detail in the documentation [15].

We note that MJOLNIR has been written specifically for multiplexing triple-axis instruments. The software is not suitable for data acquired on TAS spectrometers where only a one-dimensional sub-space of reciprocal space is measured. This was decided on the basis of the simple data structure in standard TAS experiments, where a multitude of programs are available and the complexity of data treatment is at a level reachable for most users. Further, overlaying TAS and multiplexing data are expected to be only performed on one-dimensional cuts, which can be done with conventional programs. Conversely, direct geometry time-of-flight spectrometers produce 4-dimensional data sets, whose handling is outside the scope of MJOLNIR. A multitude of software exists dealing with direct geometry time-of-flight spectrometers, including Mantid [14], Horace [12], and Dave [13].

## 2.3. Data analysis

Because all data is kept in the original unprocessed form throughout the data treatment process, different fitting algorithms and methods can be employed. This is important for non-standard Poisson and Multinomial fitting routines [27], where it is important to keep discrete neutron counts as opposed to normalised counts. Currently, fitting is supported through a series of 1D cuts, or simultaneous multifits thereof, in reciprocal space at constant energy, or constant  $Q$ . An interface to the uFit [28] program has been developed, allowing for a graphical interface for the fitting procedures. Further, down the line we also foresee an interface with other analysis software, such as SpinW [20].

## 2.4. Impact

We envision MJOLNIR to become a user-friendly standard data reduction and visualisation software for all existing and upcoming multiplexing spectrometers. This includes CAMEA@PSI,

MultiFLEX at HZB in Germany, BIFROST at the European Spallation Source in Sweden, BAMBUS at FRM2 in Germany, MARMOT at the Institut Laue–Langevin in France, and further potential instruments at the future cold neutron target station of the Spallation Neutron Source in Oakridge, USA. The software package has already been successfully used during the first commissioning phase of CAMEA@PSI in 2018 [2,29,30], and is capable of treating MultiFLEX data [31].

## 3. Program structure

The MJOLNIR package is composed of several modules, each dedicated to a specific task. A virtual model of the instrument is generated using the Geometry module. The actual data processing and visualisation objects are located in the DataSet module. The data fitting process is kept separate from the data objects in the Statistics module in an effort to clearly separate data conversion and data analysis. An advantage of this separation is the option to use different routines and software for fitting, allowing greater flexibility. As mentioned above, the fitting module has been extended with the capabilities of uFit [28] through the creation of uFit files from both the Gui and the MJOLNIR library. We refer to the documentation of uFit for details of its capabilities. General functions and repeatedly used subroutines are located in the `_tools` module. One of them is a collection of triple-axis conversion commands called `TasUBLibDEG`, which was translated from the C++ library `TasUBLib` based on Lumsden *et al.* [32]. In the following sections, we highlight some key features of the different modules.

### 3.1. Geometry

The objects and methods in the Geometry module have two principal purposes; the first is the virtual representation of the instruments, the second is to facilitate the generation of normalisation tables in which the calibrated pixel efficiency, final energy, and scattering angles of the detectors are stored. The normalisation has been separated from the implementation of the data structure such that one can generate calibration tables without knowledge of precise data structures. A virtual representation of the instrument will be particularly useful in the future, where we aim to include calculations of the resolution function and the prediction of spurious signals.

*Virtual instrument.* The creation of a virtual instrument is achieved with the `Instrument` object and the subsequent `Wedge`, `Analyser`, and `Detector` objects. These encode their real world counterparts, containing information about instrument relevant positions, directions, and  $d$ -spacings. The simulation of an instrument is implemented either via a script that adds the objects to the instrument structure, or by means of an XML file.

In the simplest case, there is a one-to-one correspondence between detectors and analysers. This is the case for MultiFLEX. The situation is more involved at CAMEA@PSI as the instrument consists of 13 position sensitive detectors inside a  $8^\circ$  wide wedge. Each detector measures neutrons scattered from 8 different analysers, typically corresponding to 24 to 64 different energies, when the prismatic analyser concept is employed. In such cases where a detector receives neutrons from different analysers, a position sensitive detector with a defined pixel numbering is simulated. This allows splitting the detectors into sub-parts, identifying neutrons from the different analysers.



```

$ MJOLNIRHistory
camea2018n000494.hdf: sc a3 30 da3 1 np 101 mn 50000 MnF2 MV=80 Ei=8.5 2t=-24 10 K
camea2018n000495.hdf: sc a3 30 da3 1 np 101 mn 50000 MnF2 MV=80 Ei=8.5 2t=-28 10 K
camea2018n000496.hdf: sc a3 100 da3 1 np 41 mn 50000 MnF2 MV=80 Ei=8.5 2t=-20 10 K
camea2018n000497.hdf: sc a3 100 da3 1 np 41 mn 50000 MnF2 MV=80 Ei=8.5 2t=-24 10 K
camea2018n000498.hdf: sc a3 40 da3 1 np 141 mn 50000 MnF2 MV=80 Ei=10 2t=-16 10 K
camea2018n000499.hdf: sc a3 50 da3 1 np 141 mn 50000 MnF2 MV=80 Ei=10 2t=-16 10 K
camea2018n000500.hdf: sc a3 50 da3 1 np 141 mn 50000 MnF2 MV=80 Ei=10 2t=-20 10 K

```

**Fig. 3.** Output of MJOLNIRHistory for data files 494 to 500 taken in 2018, showing scan command and title ("MnF2 MV=80 Ei=10 2t=-20 10 K").

### 3.2. Data

All code that is connected to the data conversion and data treatment is grouped in the Data module. This includes the DataFile and Sample objects, which refer to instrument and sample parameters of individual scan files. This, for instance, enables masking options for specified data regions in individual data files or across an entire data set.

Experiments on multiplexing instruments often produce a number of data files with similar parameters that can be combined into a single data set. For this reason we created the DataSet object that deals with multiple data files simultaneously. It represents an abstraction of a list containing DataFile objects, and hosts a number of methods and helper functions to perform cuts and plots.

**Conversion.** Once the experimental geometry and all instrument settings are known, the conversion from the detector positions to reciprocal lattice units happens in two steps. First, the  $A_4$  angles and final energies of the instrument reference frame are transformed into the instrument coordinate system, and then to the reciprocal lattice unit system of the sample. Both conversions are based on the formalism of TasUBlib [32].

Most often, a data file is created by scanning one or more instruments parameters, typically the incoming energy, the sample rotation, or the rotation angle of the analyser-detector tank that impacts the scattering angle of all detectors. MJOLNIR allows for the most general case of multiple changing parameters, which happens when scanning both the sample rotation and scattering angle simultaneously.

Data from each file is converted to a structure of shape  $n_p \times n_{\text{detectors}} \times n_{\text{pixels}}$ , where  $n_p$ ,  $n_{\text{detectors}}$  and  $n_{\text{pixels}}$  is the number of scan points, detectors and pixels per detector, respectively. One such object is created for  $A_3$ ,  $A_4$ ,  $\Delta E$ ,  $qx$ ,  $qy$ ,  $H$ ,  $K$ ,  $L$ , the normalisation, monitor and neutron count  $I$ . Here, normalisation refers to the combined analyser and detector efficiency, while the monitor count keeps track of the neutron count in the guide monitor. The converted data are the basis for any further process. We note that the normalisation, monitor and  $I$  are not combined, but kept separately.

**Performing cuts.** The direct visualisation of a complete scan file as a point cloud in 3D reciprocal space is often disadvantageous. Instead, cuts along specific lines or planes in reciprocal space, most often determined by the crystallographic high-symmetry directions, are performed to highlight specific features. Due to the usual scarcity of regions of interest it may also be preferable to avoid cuts where only low scattering areas are present. Multiple cutting and corresponding plotting methods have been implemented in MJOLNIR. These include 1D cuts along a constant energy or momentum transfer line and 2D intensity maps connecting multiple scattering points. These cuts are useful to visualise the data, and are also the basis for further data processing such as fits to the data. Further information on the visualising routine is given below.

**Masking.** As CAMEA@PSI measures a large region in reciprocal space during a single scan, additional unwanted signals are sometimes detected. These may be extrinsic features such as spurious peaks, additional phonons, or magnetic contributions that do not concern the addressed scientific question. It may be preferential to mask these contributions, before cuts or fits are performed. In MJOLNIR, this is supported by the Mask module where different options are available. They can be combined using standard Boolean algebraic operations, which are described in detail in the MJOLNIR documentation [15].

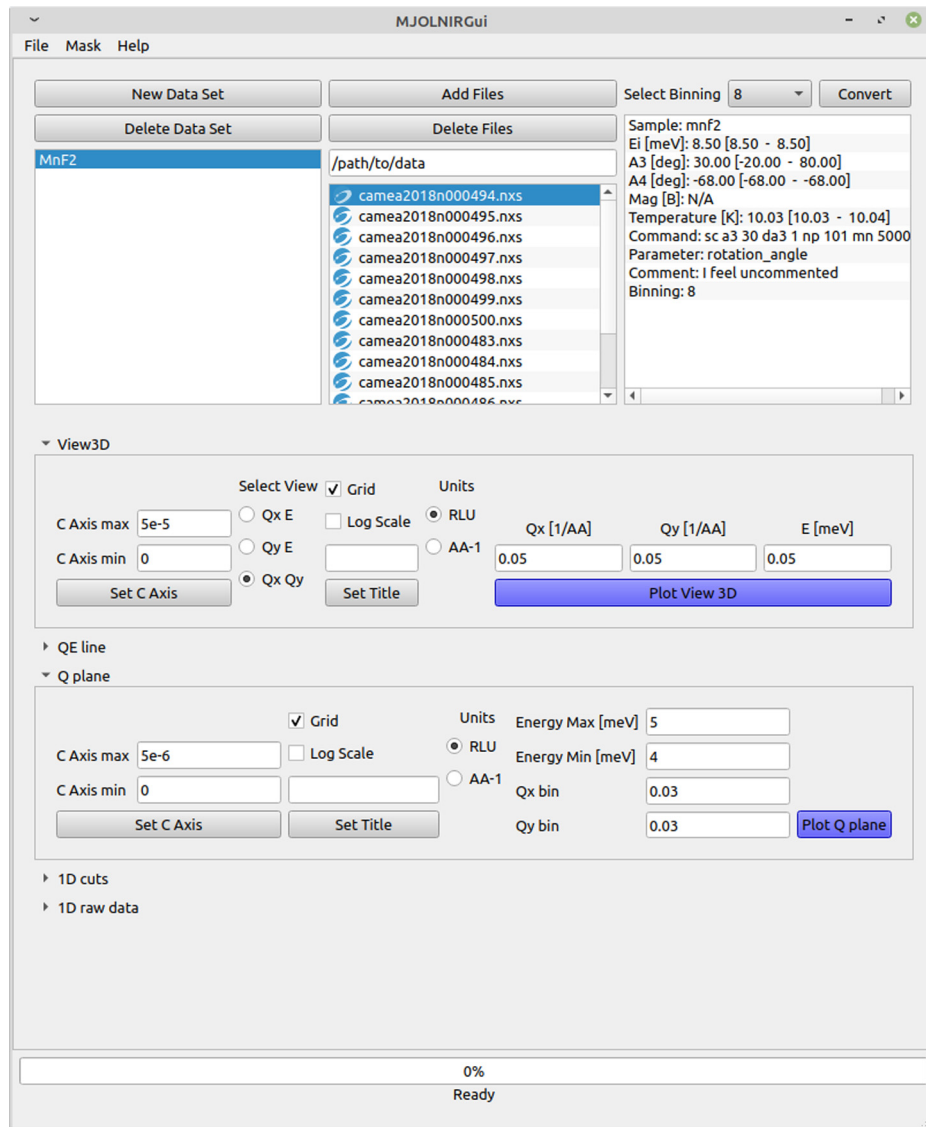
### 3.3. Statistics and fitting

The main goal of any neutron scattering experiment is to gain deeper insight into microscopic parameters of the studied material. This happens either through direct determination of microscopic properties or indirectly through a comparison with theoretical models. In both cases some kind of fitting is needed. MJOLNIR provides statistical tools, which allow for different statistical approaches. In particular when dealing with Poisson statistics, the normalisation and neutron intensity at each measured reciprocal lattice needs to be taken into account [27]. An accurate fitting routine of such data is currently developed by merging MJOLNIR with the existing software package of uFit.

### 3.4. Interfaces

Three different interfaces have been created which serve different purposes, also providing different levels of control. The main interface is the scripting interface of MJOLNIR, which is imported in either a Python script or a Jupyter notebook. It supports all features of the software and allows creating new virtual instruments, normalisation tables, and to convert and visualise the data. A limited command line interface can be used to shortcut some of the key functions. These include reporting, conversion and visualisation functions, serving as a quick way of checking data during experiments. We have also developed a graphical user interface (GUI) meant to both allow quick figure generation through a point-and-click interface, and to also guide users towards the scripting interface.

**Command line interface.** Four commands have been created that act as a short list for MJOLNIR scripts. These are the MJOLNIR3DView-method plotting the interactive View3D window, the MJOLNIRConvert-method converting data sets from HDF to NXS using a specified binning, the MJOLNIRCalibrationInspector-method that displays the current normalisation options,  $A_4$  values and final energies, and the MJOLNIRHistory-method that reports the data set content. In each case the respective data files are entered directly or through a file dialog. Further variables or input parameters can be attached to control properties such as the bin sizes used in conversion or the type of plot that is generated. An example of the MJOLNIRHistory-method is shown in Fig. 3.



**Fig. 4.** Overview of the graphical user interface of MJOLNIR denoted MJOLNIRGui, where blue coloured buttons suggest next steps of the data treatment. Selecting a data file in a data set shows its properties in the right upper-most window.

**Graphical user interface.** A user-friendly interface, MJOLNIRGui, has been created for users that prefer to avoid using Python in a scripting-based environment. The main window of MJOLNIRGui is shown in Fig. 4. The interface is built directly on top of MJOLNIR and provides access to a variety of its key features. The tool can generate different plots and inspect experimental data on the fly. We also implemented the possibility to directly generate Python scripts from the GUI interface. This is useful if the user prefers to add customary modifications to figures, for instance. Adhering to the desire of being useable across many operating systems, we wrote MJOLNIRGui using the python bindings of Qt [33]. In combination with FBS [34] it allows for the creation of installers across different operating systems.

#### 4. Using MJOLNIR

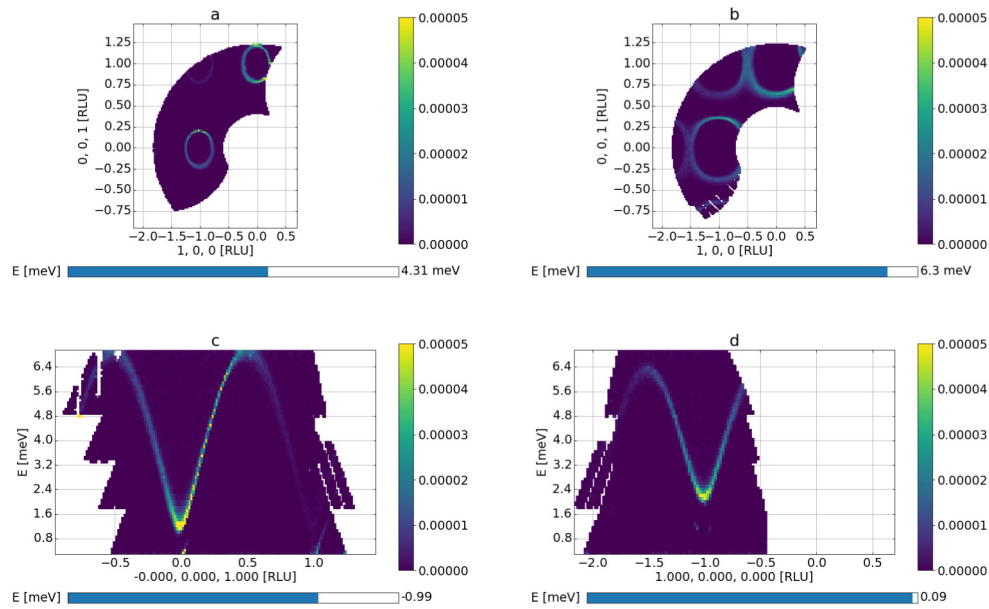
In this section we show examples of how MJOLNIR can be used. The data and a tutorial replicating the figures produced here can be found in the documentation of MJOLNIR[15]. The presented spin waves were acquired on a large MnF<sub>2</sub> single crystal at  $T = 10$  K measured at CAMEA@PSI. Further details are found in Ref. [2].

##### 4.1. Loading and converting data

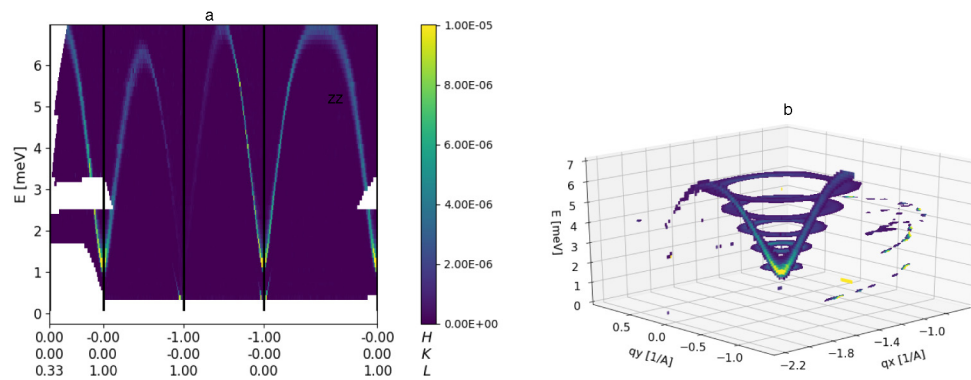
Using MJOLNIRGui, we first create a new Data set, then add the data files, load them into MJOLNIR, and finally convert the data to the reciprocal lattice space. These steps can be carried out within the GUI, or using the scripting interface. At this stage it is also possible to correct the data by adding an offset to A3 or A4, or adjusting the lattice parameters using the scripting interface.

##### 4.2. Overview of data: Viewer3D

For a quick overview of the data, the Viewer3D method is particularly useful. The method bins and plots the data in the instrument coordinate system ( $Q_x, Q_y, \Delta E$ ) or the reciprocal lattice coordinate system, ( $P_1, P_2, \Delta E$ ). The data is binned via the SciPy HistogramDD into equi-sized voxels, whose size can be specified by the user. The main reason for utilising a constant binning is to speed up the processing time. We chose to perform the binning in the orthonormal instrument coordinate system to truthfully visualise the data, which would not be the case if it was performed in a geometry where the reciprocal lattice basis vectors had different lengths. Alternatively, one could plot constant



**Fig. 5.** Series of pictures from Viewer3D showing data from  $\text{MnF}_2$  for 2 different constant- $E$  planes in Figs. a and b and two  $(q, E)$  cuts along the principal directions in Figs. c and d.



**Fig. 6.** a: Intensity plotted as function of energy and  $\mathbf{Q}$  for  $\text{MnF}_2$  using `plotCutQELine` and 5  $\mathbf{Q}$  points. b: 3D visualisation of scattering intensity for  $\text{MnF}_2$ . Transparent parts signify low intensity.

energy planes in a polar coordinate system. This is because an  $A_3$  rotation represents a rotation around the origin, which yields almost equally distributed data points as a function of  $|\mathbf{Q}|$ . We implemented this option into the `plotQPlane` method.

The `View3D` module comes with an interactive GUI that allows the user to step through the data, plane by plane, along three different directions, i.e., along the two projection vectors and the energy. It is possible to perform cuts along arbitrary directions, for which more computational power is required. Examples of the output of `View3D` are shown in Fig. 5. The script needed to produce this figure is given in the Appendix, and is generated by the Gui interface.

#### 4.3. Cuts through the data: `plotCutQELine`

It is often desirable to plot the intensity as function of energy and  $\mathbf{Q}$  along a path in reciprocal space. The `plotCutQELine`-module has been developed for this purpose. The code is composed of smaller building blocks, ensuring a successful visualisation of arbitrary two-dimensional cuts with different binning sizes and relative distances in reciprocal space.

Each cut unifies a collection of one-dimensional cuts of a certain  $\mathbf{Q}$ -width and constant energy window. An example is

shown in Fig. 6a using 5 different  $\mathbf{Q}$  points of  $\text{MnF}_2$ . A mouse-over function displays the normalised intensity, the central relative lattice position, the normalisation, monitor and the number of binning points to create the shown pixel intensity.

The plotting method also allows displaying 2D planes embedded in a three-dimensional figure as shown in Fig. 6b. The plots have been combined with constant energy plots generated by the `plotQPlane` method. The colour map is chosen such that low intensity points are transparent.

## 5. Conclusion

We have developed the `MJOLNIR` software package to convert, visualise and analyse data from multiplexing triple-axis instruments. It has already been successfully used during the commissioning of `CAMEA@PSI`, and the analysis of the collected data thereof. The package will be further extended in the future to include direct support of other multiplexing instruments, prediction of spurious, experimental planning tools, advanced fitting tools, and resolution and absorption calculations. We anticipate `MJOLNIR` to greatly benefit users of multiplexing triple-axis instruments.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

Many people have partaken in discussion concerning the software requirements and development. We would especially like to thank Christof Niedermayer and Jonas Okkels Birk for many discussion early in the development process. This laid much of the foundation for its formulation. Also Paul Steffens and Siqin Ming have participated in many fruitful discussions. We especially acknowledge them for their insights into Voronoi methods and adaptive binning. We also thank the people standing in as guinea-pigs for the first versions of the software. This includes Sofie Janas, Morten Lunn Haubro, Virgile Favre, Luc Testa, and Stephan Allenspach.

## Funding

This research was funded by the Danish Agency for Research and Innovation through DanScatt grant 7055-00007B. J.L. was supported by Nordforsk project 81695: NNSP School, Switzerland, and by the Paul Scherrer Institut, Switzerland.

HJ was supported by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 701647.

## Appendix. Viewer3D script

The following script was generated by MJOLNIRGui using the MnF<sub>2</sub> data.

```
import matplotlib.pyplot as plt
try:
    import IPython
    shell = IPython.get_ipython()
    shell.enable_matplotlib(gui='qt')
except:
    pass

from MJOLNIR import _tools
from MJOLNIR.Data import DataSet

import numpy as np
from os import path
dataFiles = _tools.fileListGenerator("483–489,494–500",r"path/to/
data",2018)

MnF2 = DataSet.DataSet(dataFiles)
# Run the converter. This automatically generates nxs-file(s).
# Binning can be changed with binning argument.
MnF2.convertDataFile(binning = 8,saveFile=False)

# Plotting data quickly in equi-sized voxels can be done by
Viewer = MnF2.View3D(0.05,0.05,0.05, grid=True)
# Above, all data is binned in voxels of size 0.05/AA, 0.05/AA, and
0.05 meV.
# Automatically, data is plotted in reciprocal lattice as provided
by the
# UB matrix saved in the data file. Alternatively, one can plot in
'raw'
# coordinates (orthogonal and in units of 1/AA) by issuing rlu=
False above.

Viewer.caxis=(0.5e-5)
# Without any intervention data is usually plotted on a useless
colour scale.
# This is countered by specifying min and max for colour by the
call above.
# Alternatively, one can provide this as input to View3D

plt.show()
```

## References

- [1] Shirane G, Shapiro SM, Tranquada JM. Neutron scattering with a triple-axis spectrometer. Cambridge; 2004.
- [2] Lass J, et al. Design and performance of the multiplexing spectrometer CAMEA. 2020, p. 1–12, [arXiv:2007.14796](https://arxiv.org/abs/2007.14796).
- [3] PSI. Rita2. 2019, <https://www.psi.ch/sinq/ritaii/description>.
- [4] Kempa M, et al. The flatcone multianalyzer setup for ILL's three-axis spectrometers. Physica B 2006;385:1080–2. <http://dx.doi.org/10.1016/j.physb.2006.05.371>.
- [5] Schmidt W, Ohl M. IN12-UFO: new frontiers for cold triple-axis spectroscopy. Physica B 2006;385–386:1073–6. <http://dx.doi.org/10.1016/j.physb.2006.05.343>.
- [6] Groitl F, et al. CAMEA - a novel multiplexing analyzer for neutron spectroscopy. Rev Sci Instrum 2016;87:035109. <http://dx.doi.org/10.1063/1.4943208>.
- [7] Sobolev O, Park JT. PUMA: Thermal three axes spectrometer. J Large-Scale Res Facil 2015;1:A13. <http://dx.doi.org/10.17815/jlsrf-1-36>, <http://jlsrf.org/index.php/lrf/article/view/36>.
- [8] Lim JA, Siemensmeyer K, Čermák P, Lake B, Schneidewind A, Inosov DS. BAMBUS: A new inelastic multiplexed neutron spectrometer for PANDA. J Phys Conf Ser 2014;592(1):012145. <http://dx.doi.org/10.1088/1742-6596/592/1/012145>.
- [9] Rodriguez JA, et al. MACS - A new high intensity cold neutron spectrometer at NIST. Meas Sci Technol 2008;19:034023.
- [10] Meng S, Toft-Petersen R, Hao L, Habicht K. Multiflexxlib: A python package for data reduction and visualization for the cold-neutron multi energy wide angle analyzer multiflexx. SoftwareX 2018;7:309–12. <http://dx.doi.org/10.1016/j.softx.2018.09.006>.
- [11] Steffens P. ILL Neutrons for society, softwares. 2018, <https://www.ill.eu/users/scientific-groups/spectroscopy/softwares/>. [Accessed 28 August 2020].
- [12] Ewings RA, et al. HORACE: Software for the analysis of data from single crystal spectroscopy experiments at time-of-flight neutron instruments. Nucl Instrum Methods A 2016;834:132–42. <http://dx.doi.org/10.1016/j.nima.2016.07.036>, [arXiv:1604.05895](https://arxiv.org/abs/1604.05895).
- [13] Azuah R, et al. DAVE: A comprehensive software suite for the reduction, visualization, and analysis of low energy neutron spectroscopic data. J Res Natl Inst Stand Technol 2009;114(6):341–58. <http://dx.doi.org/10.6028/jres.114.025>.
- [14] Arnold O, et al. Mantid - data analysis and visualization package for neutron scattering and  $\mu$ sr experiments. Nucl Instrum Methods A 2014;764:156–66. <http://dx.doi.org/10.1016/j.nima.2014.07.029>.
- [15] PSI. Data treatment - CAMEA - PSI. 2020, <https://www.psi.ch/en/sinq/camea/data-treatment>. [Accessed 06 August 2020].
- [16] Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng 2007;9(3):90–5. <http://dx.doi.org/10.1109/MCSE.2007.55>.
- [17] Virtanen P, et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. Nature Methods 2020;17:261–72. <http://dx.doi.org/10.1038/s41592-019-0686-2>.
- [18] The pandas development team. pandas-dev/pandas: Pandas. Zenodo; 2020, <http://dx.doi.org/10.5281/zenodo.3509134>.
- [19] MATLAB. 9.7.0.1190202 (R2019b). Natick, Massachusetts: The MathWorks Inc.; 2018.
- [20] Toth S, Lake B. Linear spin wave theory for single-Q incommensurate magnetic structures. J Phys Condens Matter 2015;27(16):166002. <http://dx.doi.org/10.1088/0953-8984/27/16/166002>.
- [21] Andersen KH, et al. The instrument suite of the ESS. Nucl Instrum Methods A 2020;957:163402.
- [22] Eckold G, Sobolev O. Analytical approach to the 4d-resolution function of three axes neutron spectrometers with focussing monochromators and analysers. Nucl Instrum Methods A 2014;752:54–64. <http://dx.doi.org/10.1016/j.nima.2014.03.019>.
- [23] Groitl F, et al. MULTIFLEX - the new multi-analyzer at the cold triple-axis spectrometer FLEXX. Sci Rep 2017;7(1):1–12. <http://dx.doi.org/10.1038/s41598-017-14046-z>.
- [24] Birk JO, et al. Prismatic analyser concept for neutron spectrometers. Rev Sci Instrum 2014;85(11):113908. <http://dx.doi.org/10.1063/1.4901160>, [arXiv:1408.0551](https://arxiv.org/abs/1408.0551).
- [25] ESS. BIFROST ESS. 2020, <https://europeanspallationsource.se/instruments/bifrost>. [Accessed 23 September 2020].
- [26] Freeman P, et al. CAMEA ESS - the continuous angle multi-energy analysis indirect geometry spectrometer for the ESS. EPJ Web Conf 2015;83:03005.
- [27] Lass J, Bøggild ME, Hedegård P, Lefmann K. Multinomial, Poisson and Gaussian statistics in count data analysis. J Neutron Res 2020;1:1–24. <http://dx.doi.org/10.3233/jnr-190145>, [arXiv:2006.03812](https://arxiv.org/abs/2006.03812).
- [28] Brandl G. Ufit, MLZ - wiki. 2018, <https://wiki.mlz-garching.de/ufit:index>. [Accessed 04 June 2020].
- [29] Janas S, et al. Classical spin liquid or extended critical range in h-YMnO<sub>3</sub>? 2020, [arXiv:2006.03396](https://arxiv.org/abs/2006.03396).



- [30] Allenspach S, et al. Three dimensional criticality in the quasi two-dimensional quantum magnet  $\text{Ba}_{0.9}\text{Sr}_{0.1}\text{CuSi}_2\text{O}_6$ . 2020, in preparation.
- [31] Lass J, Andersen CR, Leerberg HK, Birkemose S, Toth S, Stuhr U, et al. Field-induced magnetic incommensurability in multiferroic  $\text{Ni}_3\text{TeO}_6$ . *Phys Rev B* 2020;101(5):54415. <http://dx.doi.org/10.1103/PhysRevB.101.054415>, [arXiv:1909.13734](https://arxiv.org/abs/1909.13734).
- [32] Lumsden MD, Robertson JL, Yethiraj M. UB Matrix implementation for inelastic neutron scattering experiments. *J Appl Crystallogr* 2005;38(3):405–11. <http://dx.doi.org/10.1107/S0021889805004875>.
- [33] Company TQ. Qt, Cross-platform software development for embedded & desktop. 2020, <https://www.qt.io/>. [Accessed 04 May 2020].
- [34] Herrmann M. FBS web-page. 2020, <https://build-system.fman.io/>. [Accessed 04 May 2020].