Journal of Physics Communications

PAPER • OPEN ACCESS

The software package MIEZEPY for the reduction of MIEZE data

To cite this article: Alexander Schober *et al* 2019 *J. Phys. Commun.* **3** 103001

View the article online for updates and enhancements.

Recent citations

- Neutron MIEZE spectroscopy with focal length tuning
  J K Jochum *et al*

This content was downloaded from IP address 129.187.254.46 on 15/04/2020 at 20:01

Journal of Physics Communications

OPEN ACCESS

RECEIVED
31 July 2019

REVISED
17 September 2019

ACCEPTED FOR PUBLICATION
30 September 2019

PUBLISHED
14 October 2019

Original content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

The software package MIEZEPY for the reduction of MIEZE data

Alexander Schober[1], Andreas Wendl[2], Franz X Haslbeck[2,3], Johanna K Jochum[4], Leonie Spitz[4] and Christian Franz[4,5]

[1] Jülich Centre for Neutron Science (JCNS) at Heinz Maier-Leibnitz Zentrum (MLZ) Forschungszentrum Jülich GmbH Lichtenbergstr. 1 85748 Garching, Germany
[2] Physik Department, Technische Universität München, D-85748 Garching, Germany
[3] Institute for Advanced Study, Technische Universität München, D-85748 Garching, Germany
[4] Heinz Maier-Leibnitz Zentrum (MLZ), Technische Universität München, D-85748 Garching, Germany
[5] Author to whom any correspondence should be addressed.

E-mail: a.schober@fz-juelich.de, andreas.wendl@frm2.tum.de, franz.haslbeck@frm2.tum.de, johanna.jochum@frm2.tum.de, leonie.spitz@frm2.tum.de and christian.franz@frm2.tum.de

Keywords: data reduction, MIEZE, NRSE, Python, PyQt

Abstract
Modulation of intensity with zero effort (MIEZE) is a neutron resonant spin echo technique which allows to measure the intermediate scattering function $S(Q,\tau)$ under depolarizing conditions. Since MIEZE produces a complex four dimensional dataset, we have developed the software package MIEZEPY to reduce the dataset and extract $S(Q,\tau)$ in a user friendly manner. This is an essential step in establishing the MIEZE technique and improving user operation. MIEZEPY was written in Python as an open source package and was developed on GitHub. In this paper the framework and implementation of this package as well as the physical and mathematical principles underlying the data reduction procedure will be introduced to lay out the complexity of this task.

1. Introduction

Neutron Spin Echo (NSE) [1] is a Larmor labelling technique that was developed to study slow dynamics in liquid solutions, polymers and spin glasses by directly measuring the intermediate scattering function $S(Q,\tau)$. For NSE and all other spin echo techniques the energy resolution is decoupled from the neutron wavelength spread and instead encoded in the beam polarization to achieve highest energy resolution. Neutron resonance spin echo (NRSE) [2, 3], a variation of classical NSE, replaces the long solenoids which generate the precession fields with compact resonant spin flippers. This allows to adapt the technique for various science cases [4–6]. Modulation of IntEnsity with Zero Effort (MIEZE), being such a modification, is insensitive to depolarizing (e.g. magnetic) samples or depolarizing sample environments, such as large magnetic fields. MIEZE is furthermore very well-suited for incoherently scattering (e.g. hydrogen containing) samples since it does not produce the 2/3 background originating from spin flip scattering that is present for conventional NSE/NRSE. In essence, MIEZE is a high-resolution, spin-echo, time-of-flight (TOF) technique where all beam preparation is performed in front of the sample. Typical applications for MIEZE are the investigation of quantum phase transitions, superconductors, vortex lattices, skyrmions, ferromagnetic materials and hydrogen containing samples [7]. The MIEZE technique is implemented at the RESEDA [8, 9] beamline at the Heinz Maier-Leibnitz Zentrum (MLZ), a reactor source in Germany, the VIN-ROSE beamline at the spallation source J-PARC, Japan [10], as well as at the LARMOR instrument at ISIS, United Kingdom [11].

In a MIEZE experiment the intermediate scattering function $S(Q,\tau)$ is measured for different Fourier times τ by varying the modulation frequency $2\Delta\omega$ of the sinusoidally intensity modulated neutron beam, where $\Delta\omega = \omega_b - \omega_a$ is the frequency difference of the first and second resonant flipper. The Fourier time τ is directly proportional to the modulation frequency. During a measurement the detector samples each period of the signal with 16 time channels and repeatedly sums up the counts in each time channel over many periods of the sine. The readout frequency of the detector is phase locked to the modulation frequency to assure a stable assignment to the time channels during the integration time. This process happens for every detector pixel leading to a

© 2019 The Author(s). Published by IOP Publishing Ltd

complex data structure. The software package MIEZEPY presented here allows to reduce such datasets into a user friendly format and extract the intermediate scattering function $S(Q, \tau)$.

2. MIEZE data reduction

In the following section the process of data reduction will be explained to elucidate the complexity of the procedure and the challenges it poses for a user friendly data reduction software. Typically for MIEZE, 2d position sensitive detectors are employed to achieve high spatial resolution. In addition to the necessary time resolution, this leads to a 3d dataset. To increase detector efficiency, some MIEZE detectors contain several consecutive detection foils adding another dimension to the dataset.

The MIEZEPY software was developed at the MLZ based on datasets produced by the resonant neutron spin echo spectrometer RESEDA. RESEDA uses a CASCADE detector [12] with 8 detection foils where each data file has the dimensions: $128 \cdot 128$ (pixel) $\cdot$ 16 (time channels) $\cdot$ 8 (detector foils) leading to data files with a size of 8 MB per Fourier time.

In contrast to NSE, the intensity of the MIEZE signal varies in time. Each time channel measures the intensity

$$I_{tc} = \frac{I_0}{\Delta\phi}\int_{-\Delta\phi/2}^{\Delta\phi/2} \sin(\phi - \phi_0)d\phi = I_0 \sin(\phi_0)\frac{\sin \Delta\phi/2}{\Delta\phi/2}, \tag{1}$$

for an arbitrary sinusoidal intensity, with a phase $\phi_0 = \Delta\phi \cdot tc$, which depends on the number of the time channel tc and the width of each time channel $\Delta\phi = 2\pi/\#time\ channels$. The contrast reduction is then given by the sinc-function in equation (1) and amounts to 0.64% for 16 time channels. Since this factor applies for the datasets as well as the resolution measurement, it cancels in normalization and does not influence the intermediate scattering function.

Although in the following the process is described on the basis of the spectrometer RESEDA, the software can be used for any number of foils, number of time channels and spatial resolution of the detector.

Figure 1 explains the data reduction schematically. Starting from the full four-dimensional dataset for one Fourier time (1(a)), the different detection foils are initially corrected separately (steps (b) through (e) in figure 1) before the information of different foils can be combined. Firstly, phase corrections due to flight path differences and foil distortions (see section 3 for more details) are applied to every pixel on each foil separately (1(b)→(c)). Afterwards, the sinusoidal signal is in phase in every pixel and a region of interest (ROI) can be chosen (1(d)). Then the intensities of all pixels within the ROI and the selected foils are summed up for each of the 16 time channels (1(d)→(e)).

The resulting intensity $I(t_{tc}, \Delta\omega)$, where t_{tc} is the discrete time channel, is considered to follow a Poisson statistic with the error $\delta_I(t_{tc}, \Delta\omega) = \sqrt{I(t_{tc}, \Delta\omega)}$. This $I(t_{tc}, \Delta\omega)$ is then fit with a sine function $f(t_{tc}, \Delta\omega) = a(\Delta\omega) \cdot \sin(2\pi \cdot t_{tc}/16 + \phi(\Delta\omega)) + m(\Delta\omega)$, where a is the amplitude, ϕ is the phase shift and m is the median offset with their respective fit errors $\delta_a(\Delta\omega)$, $\delta_\phi(\Delta\omega)$ and $\delta_m(\Delta\omega)$. Both a and m (and their respective error) are normalized over the neutron monitor to avoid reactor flux dependent errors leading to a_m and m_m with the respective errors $\delta_{a_m}(\Delta\omega)$ and $\delta_{m_m}(\Delta\omega)$. Assuming that a background measurement was performed with $a_m^B(\Delta\omega)$, $m_m^B(\Delta\omega)$, $\delta_{a_m^B}(\Delta\omega)$ and $\delta_{m_m^B}(\Delta\omega)$ the evaluated amplitude, median offset and associated errors respectively, the contrast $C(\Delta\omega)$ can be written as:

$$C(\Delta\omega) = \frac{a_m(\Delta\omega) - m_m^B(\Delta\omega)}{m_m(\Delta\omega) - m_m^B(\Delta\omega)}, \tag{2}$$

with the associated error on the contrast as:

$$\delta_C(\Delta\omega) = \left[\left(\delta_{a_m}(\Delta\omega)\frac{\partial C(\Delta\omega)}{\partial a_m(\Delta\omega)}\right)^2 + \left(\delta_{a_m^B}(\Delta\omega)\frac{\partial C(\Delta\omega)}{\partial a_m^B(\Delta\omega)}\right)^2 + \left(\delta_{m_m}(\Delta\omega)\frac{\partial C(\Delta\omega)}{\partial m_m(\Delta\omega)}\right)^2 + \left(\delta_{m_m^B}(\Delta\omega)\frac{\partial C(\Delta\omega)}{\partial m_m^B(\Delta\omega)}\right)^2\right]^{\frac{1}{2}} \tag{3}$$

Note that in the absence of a background measurement, all background related values can be set to zero to yield the contrast and the associated error.

The Fourier time τ may be calculated from instrument parameters:

$$\tau_{mieze}(\Delta\omega) = \frac{2m_n^2}{h^2} \cdot \lambda^3 \cdot \Delta\omega \cdot L_{sd}, \tag{4}$$

with m_n and λ the neutron mass and wavelength and L_{sd} the sample detector distance. The scattering vector Q may be calculated from the scattering angle 2Θ which is composed of the angle of the secondary spectrometer

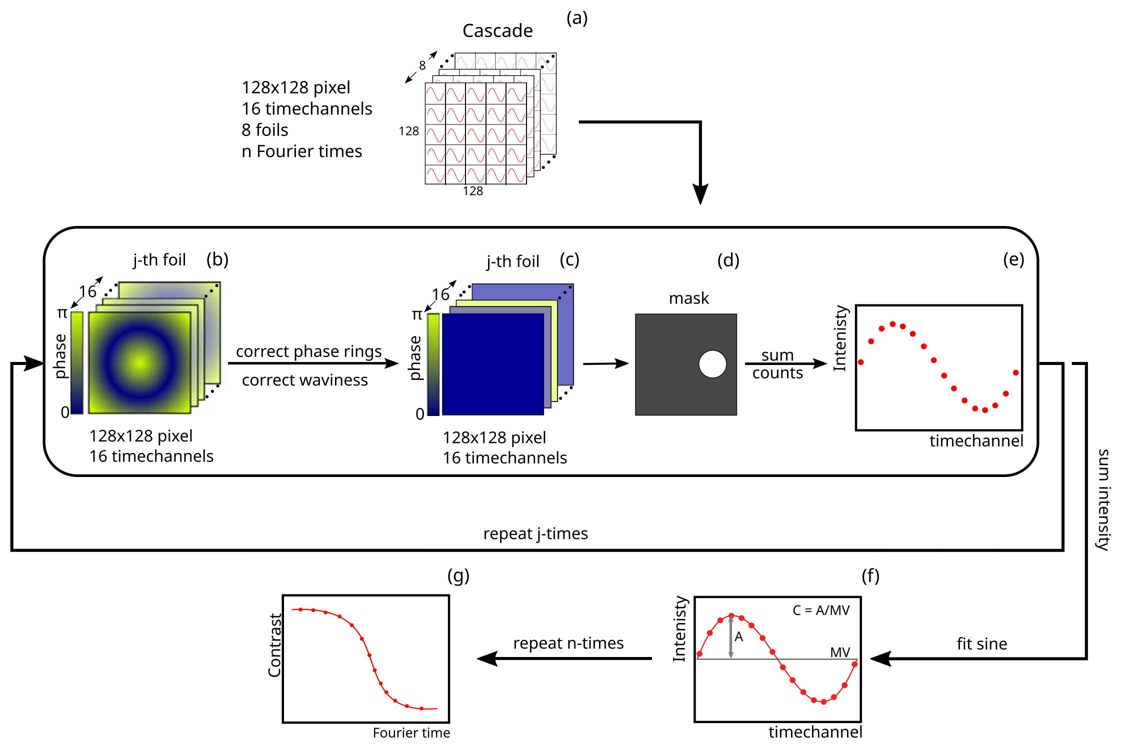

Figure 1. Schematic representation of the data reduction process. (a) The four-dimensional dataset generated by the CASCADE neutron detector comprises 128 ·128 (pixel) ·16 (time channels) ·8 (detector foils) per instrument setting (Fourier time). (b)–(c) The raw data for one detector foil is corrected for phase rings and foil distortions. (d) A region of interest (ROI) is selected and (e) counts from all pixels in the ROI are summed up. Steps (b)–(e) are repeated for all detector foils where the MIEZE signal is visible. The counts of the selected foils are summed up time channel wise. (f) A sine is fitted to the data and the contrast C is calculated for this Fourier time τ. (g) Contrast values for various Fourier times yield the reduced spectrum.

arm and the position of the ROI on the detector. As a result it is possible to write $C(\Delta\omega)$ as $C(\tau_{mieze})$. The contrast depending solely on the Fourier time can be identified with the intermediate scattering function $S(Q, \tau)$ under the assumption of small energy transfers in comparison with the incident neutron energy, i.e. where the spin echo assumption holds [13]. In the case of large energy transfers a numerical calculation needs to be used. This procedure is described elsewhere [14] and will later be added to MIEZEPY.

3. Correction of detector effects

Two effects cause flight path differences from the sample to the detector and therefore phase differences. On the one hand, typical MIEZE detectors are flat leading to concentric phase rings around the detector center as depicted in figure 2(a). On the other hand, the detector surface might be distorted. This foil height profile results in an additional phase shift depending on the individual detector and detector foil. A particularly drastic example is shown in figure 2(b). These effects lead to a phase difference between two pixels on the detector which can be written as

$$\Delta\phi(\Delta L_{SD}, \Delta\omega, v) = \Delta L_{SD} \cdot 2\Delta\omega/v, \qquad (5)$$

where ΔL_{SD} corresponds to the flight path difference and v to the neutron velocity [15, 16]. This difference is visualized in figure 2(c1) for two pixels at different distances to the detector center. This effect depends on the used Fourier time and becomes significant for $\tau \gtrsim 0.1$ ns. Summing the intensities of various pixels in a ROI with different phases, the resulting contrast is reduced. In figure 2(c2) the sinusoidal intensity as function of time is shown for pixel one and two from panel (b). The contrast of the sum of these two intensities is lower than the ideal case.

To correct for the phase rings and the foil distortions, we measure the height profile of the individual detector foils. For this, the detector is illuminated homogeneously using an elastic scatterer at a suitable modulation frequency, chosen in a way that phase differences are close to 2π over the whole detector. The sinusoidal signal is fitted for every individual pixel to determine the phase in each of them. Using equation (5) the distance ΔL_{SD} of the pixel from the ideal position may be extracted. The distance shift due to the flight path from the sample (figure 2(a)) needs to be subtracted to obtain the foil height profile. Using this profile, the phase can be calculated for any modulation frequency and thus a correction can be applied to any Fourier time using

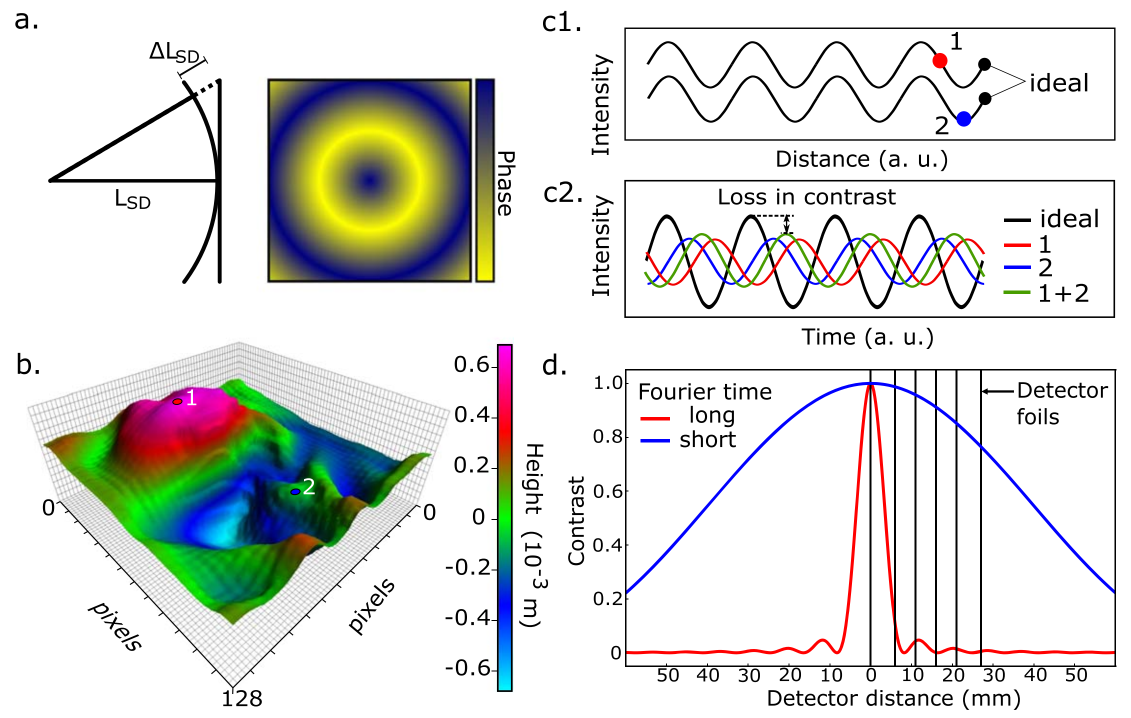

Figure 2. (a) Schematic visualization of how differences in the flight path create phase-rings on a flat detector. (b) Plot of the distortion of one of the RESEDA detector foils. (c) Schematic explanation of the difference in the spatial acquisition which induces a shift of the temporal sine function and thus a reduction in the overall contrast. (d) Schematic representation of the size of the MIEZE signal in space with respect to the detection foils

equation (5). This calculated phase shift is then translated into a time channel index shift of

$$n = (\Delta\phi \cdot 16/2\pi)\%16 \tag{6}$$

for 16 time channels. For every pixel all time channel indices are shifted by this amount n. Since every detector foil has a different distortion, the procedure needs to be repeated for every foil. In principle, this method can be incorporated into the instrument control software to correct data already upon recording.

Alternatively, the resolution measurement recorded for every Fourier time can be used to determine the relative phase shift of the individual pixels. With this phase shift the sample data is then corrected as described above. The disadvantage of this method is that very high statistics are required for each resolution measurement compared to the aforementioned determination of the height profile which is performed only once for the detector.

After the phase correction is performed, a possible error in phase may still remain since the phase shift can only be performed in the discrete steps of the time bins. The maximum deviation for one pixel is equivalent to one time channel, however a ROI contains >100 pixels. For a constant probability of phase shifts throughout all pixels, the error is 0.648%.

The phase correction is successful if the phase differences after the correction are smaller than a time channel. Rearranging equation (5) we find the limit for the phase precision

$$\delta\phi = \sigma_{\Delta L_{SD},el}\frac{2\Delta\omega_{data}}{v_{data}} = \sigma_{\phi_{el}}\frac{\Delta\omega_{data}\lambda_{data}}{\Delta\omega_{el}\lambda_{el}} < 2\pi/16 \tag{7}$$

for a dataset with wavelength λ_{data} and frequency difference $\Delta\omega_{data}$. $\sigma_{\Delta L_{SD},el}$ is the error of the foil distortion which is determined from the error of the phase $\sigma_{\phi_{el}}$ in the elastic measurement and the corresponding frequency difference $\Delta\omega_{el}$ and wavelength λ_{el}. Hereby it is assumed that the uncertainties in the sample detector distance, frequencies and wavelengths can be neglected, since they are very small and would lead to a constant phase offset in all pixels and therefore not to phase differences.

MIEZE is a spin-echo technique and the signal is only observable in a finite region around the spin-echo point where the detector is placed. The spatial width of the MIEZE signal decreases with increasing modulation frequency $2\Delta\omega$ as described in more detail in [13, 17]. At RESEDA, a detector with eight consecutive foils is employed to increase the detector efficiency [12]. At short Fourier times, the signal is observable on all detector foils whereas for highest Fourier times it is detectable only on one foil as shown on figure 2(d). Detector foils with an insufficient contrast are excluded from data reduction.

4. Software architecture

A MIEZE data reduction software has to fulfill several requirements. The used programming language should be readable for common scientific staff and still provide enough flexibility to implement complex algorithms and multi-processing. Furthermore, a GUI solution that satisfies the modern model/view/controller architecture is desirable [18]. Finally, the implementation on different platforms should be effortless.

Python [19] was chosen for MIEZEPY as it is the third most popular programming language [20] and is widely used within the scientific community [21]. Python is open-source and distributed under the GNU General Public License (GPL). It offers a multitude of scientific packages like numpy [22] and scipy [23] and is supported on Windows, MacOs and Linux. While multiple GUI packages are available to design and implement GUIs in Python, PyQt [24] was chosen because it offers a comparable visual design across all platforms PyQt consists of bindings exposing all objects of *The Qt Company*'s Qt application framework [25] to Python. Qt has a highly maintained documentation and GUI design tools, thus, allowing a rapid design and implementation of the GUI. It also supports model/view/controller architecture [18], bringing MIEZEPY in line with a modern software development philosophy. Choosing PyQt allows the use of the open source SimplePlot [26] package (based on PyQtGraph [27]) for data visualization. It offers both efficient and interactive visualization of 2d and 3d data.

4.1. Organization

The structure of the software needs to be optimized to perform specific tasks while facilitating maintainability and testing. Therefore, the core functionality, such as the phase correction or reduction procedure, is separated from the GUI logic. They are separated into two folders:

- *Core*: encapsulating the I/O, data management, reduction and fitting.
- *GUI* : user interface framework and business logic including all the dialogues, buttons and visualizations.

A schematic representation of the software architecture is presented in figure 4(a). No module declared within the *Core* should at any point include (import in the case of Python) a module from the GUI. Scripts for phase correction and reduction can be accessed via the GUI to gain flexibility. The modules in the *Core* should also include tests that can be implemented via continuous implementation tools. The next subsections treat the data organization as well as the phase correction and contrast calculation in detail.

4.1.1. The Core: Implementation of the phase correction

The phase correction consists of two methods explained in section 3. The two methods differ only in how they calculate the correction pattern while they use the same correction procedure on the dataset.

The *height profile* procedure uses *a priori* knowledge from the neutron flight path to individual detector pixels combined with an established height profile of the detector foils. Starting from the uncorrected data shown in figure 3(a), the MIEZEPY package proceeds as follows:

(i) The height profile of the detector is loaded as a three dimensional array consisting of the foil number, the x and y pixels. A representation of the foil distortion can be seen in figure 3(d).

(ii) In a second step the expected phase on each pixel of the detector is calculated using only the neutron flight path variation for each Fourier time as shown in figure 3(e).

(iii) A four dimensional array representing the phase shift induced by the flight path and foil height variation is created for each foil and Fourier time at each pixel position. Since the sinusoidal signal is discretized in the dataset due to the 16 time channels, the phase shift needs to be discretized accordingly (c.f. figure 3(b)). This map is called the phase correction pattern which indicates by how many time channels each pixel needs to be shifted to obtain the highest phase coherence on the foil.

This process assumes that the foil distortion did not change between the high exposure acquisition and the current measurement. When the height profile measurement is either outdated or not present, it is possible to use the resolution measurement of the current measurement to perform the phase correction instead:

(i) At first, the resolution measurement is fitted to extract the phase on each pixel and foil. If the intensity in one pixel is insufficient, neighboring pixels are combined.

(ii) The phases are discretized according to the 16 time channels and the phase correction pattern is created.

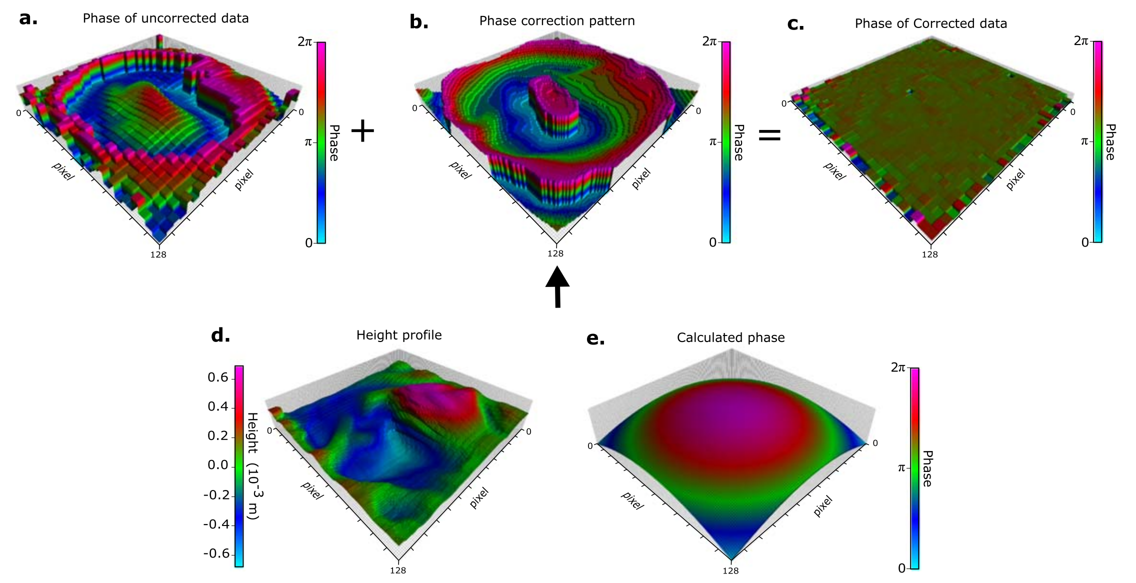

Figure 3. (a) The measured phase on a given detector foil, Fourier time and time channel. (b) Representation of the correction pattern obtained from a *height profile* procedure. (c) Corrected phase from (a) using the correction pattern of (b). (d) Height profile used for the *height profile* procedure. (e) Calculated phase shift induced by flight path variations used for the *height profile* procedure.

(iii) This process has to be repeated for each Fourier time individually.

Both processes above result in a phase correction pattern for each Fourier time that has to be applied to the dataset. Each time channel is shifted according to the phase correction pattern. Given the uncorrected phase in figure 3(a) and the correction pattern in figure 3(b), the coherent phase representation in figure 3(c) is obtained.

4.1.2. The Core: Implementation of the contrast calculation

Once the phase has been corrected, the signal in each pixel on each detector foil has the same phase and, therefore, the intensities can be summed over a region of interest. The contrast is the amplitude divided by the mean value of the sinusoidal fit of the data where each detector foil is summed up.

(i) The contrast is determined for each Fourier time of the dataset as well as for the resolution measurement.

(ii) Dividing the contrast of the dataset by the contrast of the resolution measurement, the resolution effects of the spectrometer are corrected and the intermediate scattering function $S(Q, \tau)$ is obtained.

(iii) As an initial guess, the intermediate scattering function (figure 1(g)) for each parameter is fitted by a simple exponential decrease.

4.1.3. The GUI: An intuitive work-flow

The GUI in MIEZEPY tries to follow a philosophy where the work-flow of reducing MIEZE datasets is self explanatory. A panel representation was chosen for MIEZEPY to avoid many overlapping windows. Each panel that can be selected from the left hand side navigator represents a distinct step in the reduction process. They are to be used consecutively from the top to the bottom. The six thematic panels are organized as follows:

- Environment management panel: In MIEZEPY, each measurement series which can be normalized with the same resolution curve can be grouped in one environment. Each environment contains exactly one resolution measurement and all corresponding datasets. The associated panel serves mostly the management of loaded environments.
- Data management panel: Here, the 4d datasets at a given echo time for a given parameter such as temperature or magnetic field, are selected, visualized and loaded into the correct environment. This can be seen in figure 4(b).
- Mask management panel: In this panel, masks can be created and visualized. The masks can be used either for the phase correction using the resolution measurement or as ROIs in the data reduction.
- Phase correction and data reduction panel: Here, the parameters for data reduction are chosen such as the background and resolution measurement, the method for phase correction and which foils to consider. The

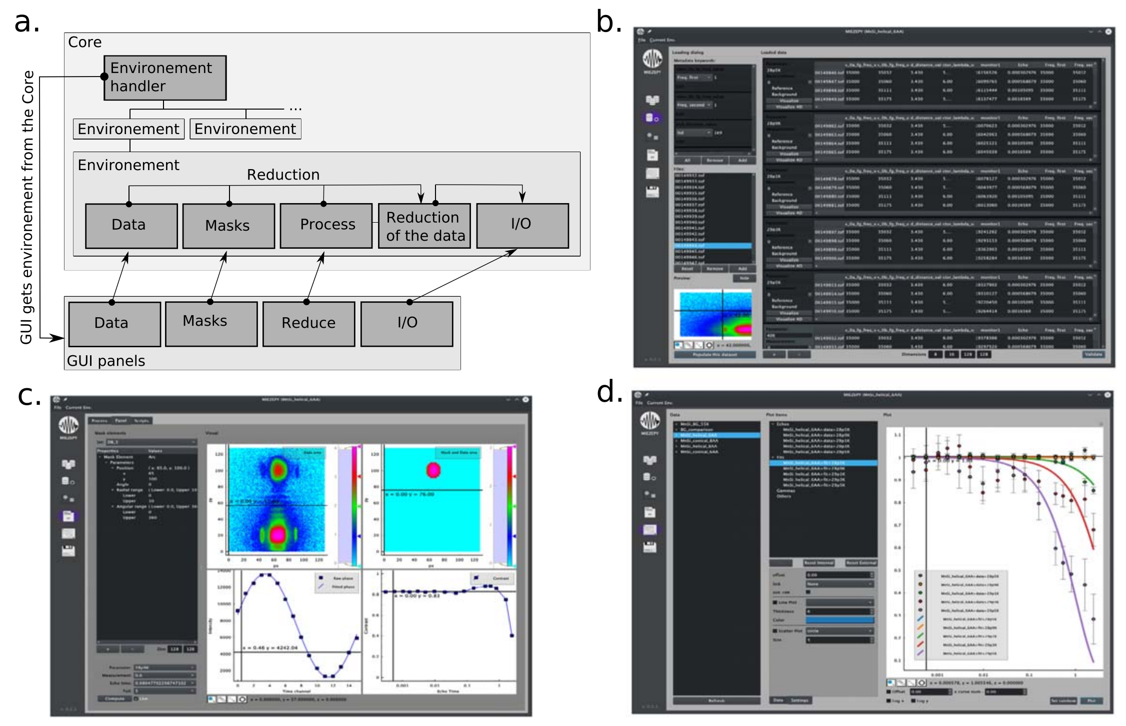

Figure 4. (a) Schematic representation of the software architecture. *Core* and *GUI* are separated. The latter calls upon the former to perform the correction and reduction tasks. (b) Example of the data panel with loaded data. (c) The phase correction and reduction panel showing the quick view tab. (d) The intermediate scattering function $S(Q, \tau)$ obtained from the dataset in (b) as depicted in the visualization panel.

data reduction is organized in scripts which are filled and started by the GUI. Furthermore, the scripts can be edited directly to retain flexibility. The panel contains a quick view of the data that allows preliminary insights on what the reduction process might yield. This can be seen in figure 4(c).

- Reduction result visualization panel: After the contrast calculation in the previous step, the data reduction results from different environments can be visualized. The data can be exported as a structured text file or an image file. An example is shown in figure 4(d).
- Input output (I/O) panel: The saving of an environment writes the masks, scripts and organization of the data files to the hard disk, where the latter symbolically points to the current location of the data. Loading these files generates the described environment. This allows to share the project easily between different scientists and creates a detailed documentation of the data reduction process.

An example for an experiment which was evaluated with this software is the investigation of fluctuation skyrmion textures in MnSi, shown in the screen-shots in figures 4(b)–(d). A detailled discussion of this example will be part of an upcoming publication [28].

4.2. Availability, documentation and continuous integration

The software package is written in Python and licensed under GNU General Public License v2.0. The copyright belongs to JCNS (Jülich Center for Neutron Scattering) and the the TUM (Technische Universität München). The package is freely accessible under its GitHub repository at https://github.com/scgmlz/NSE_Soft and can be forked and modified. Specific modifications or improvements can be requested either through an issue creation on GitHub or by contacting the maintainers by email.

A documentation is currently under development and is accessible online at https://scgmlz.github.io/MIEZEPY_website/. It is supposed to reach maturity throughout the end of 2019.

5. Conclusion

We have presented the software package MIEZEPY, a user-friendly tool with graphical interface for the data reduction of MIEZE data. MIEZEPY is written in Python and PyQt and developed on GitHub as an open source package. The purpose of MIEZEPY is the reduction of the complex 4d MIEZE dataset and extraction of the intermediate scattering function $S(Q, \tau)$. The software is capable of correcting the phase rings that appear at high

Fourier times as a consequence of the flat detector as well as foil distortions with two methods, namely the height profile method and the resolution method. Furthermore, all detector foils in case of the CASCADE detector can be automatically summed up to make use of the full detector efficiency. The reduced data can be exported into a structured text file for further analysis by the user. In addition, the software offers to fit $S(Q, \tau)$ with a simple exponential decay as an initial guess.

In the future, more complex fitting procedures will be available, once such a feature is included in SimplePlot. This will enhance the capabilities of MIEZEPY towards a data analysis tool. An additional future development includes the non-locking of the GUI through the phase correction and contrast calculation, which can be achieved by moving these processes to different threads. At this stage of maturity we expect to reach version 1.0.0 of MIEZEPY. Furthermore, it is desirable to include the correction procedures used in and developed for MIEZEPY also in the NICOS instrument control software. This would allow for an online preliminary data analysis and a better user experience.

Acknowledgments

We wish to thank S Säubert and L Beddrich from the Heinz Maier-Leibnitz Zentrum for useful discussions and M Köhli from U Heidelberg for support and discussion of the CASCADE detector system. Financial support through the BMBF project 'Longitudinale Resonante Neutronen Spin-Echo Spektroskopie mit Extremer Energie-Auflösung' (Förderkennzeichen 05K16WO6) is gratefully acknowledged. This work was supported by the German Research Foundation (DFG) and the Technical University of Munich within the Open Access Publishing Fund. The work by F X H was supported through the Hans Fischer fellowship of M Janoschek of the Technische Universität München Institute for Advanced Study, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under Grant agreement No. 291763.

ORCID iDs

Johanna K Jochum https://orcid.org/0000-0002-0066-0944
Christian Franz https://orcid.org/0000-0001-6820-2774

References

[1] Mezei F 1972 *Zeitschrift für Physik A—Hadrons & Nuclei* **255** 146–60
[2] Golub R and Gähler R 1987 *Phys. Lett.* A **123** 43–8
[3] Häussler W and Schmidt U 2005 *Phys. Chem. Chem. Phys.* **7** 1245–9
[4] Keller T, Habicht K, Klann H, Ohl M, Schneider H and Keimer B 2002 *Applied Physics A–Materials & Science Processing* **74** S332–5
[5] Groitl F, Keller T, Quintero-Castro D L and Habicht K 2015 *Rev. Sci. Instrum.* **86** 025110
[6] Rekveldt M H and Kraan W H 1999 *J. Neutron Res.* **8** 53–70
[7] Franz C *et al* 2019 *J. Phys. Soc. Jpn.* **88** 081002
[8] Maier H, Zentrum L *et al* 2015 RESEDA: Resonance spin echo spectrometer. *Journal of Large-Scale Research Facilities* **A14** 1–3
[9] Franz C, Soltwedel O, Fuchs C, Säubert S, Haslbeck F, Wendl A, Jochum J K, Böni P and Pfleiderer C 2019 *Nucl. Instrum. Methods Phys. Res., Sect. A* **939** 22–9
[10] Endo H, Oda T, Hino M and Hosobata T 2019 *Physica B Condensed Matter* **564** 91–3
[11] Plomp J 2019 *Weblog: First MIEZE signals generated by LARMOR* (https://larmor.weblog.tudelft.nl/2019/01/06/first-mieze-signals-generated-by-larmor/)
[12] Köhli M, Klein M, Allmendinger F, Perrevoort A K, Schröder T, Martin N, Schmidt C J and Schmidt U 2016 *J. Phys. Conf. Ser.* **746** 012003
[13] Keller T, Golub R and Gähler R 2002 Neutron spin echo—a technique for high-resolution neutron scattering *Scattering* ed R Pike and P Sabatier (New York: Academic) pp 1264–86
[14] Franz C, Soltwedel O, Säubert S, Wendl A, Gottwald W, Haslbeck F X, Spitz L and Pfleiderer C 2019 in Review
[15] Martin N 2018 *Nucl. Instrum. Methods Phys. Res., Sect. A* **882** 11
[16] Brandl G, Georgii R, Häußler W, Mühlbauer S and Böni P 2011 *Nucl. Instrum. Methods Phys. Res., Sect. A* **654** 394
[17] Jochum J K, Wendl A, Keller T and Franz C 2019 in preparation
[18] Reenskaug T M H 2003 (http://home.ifi.uio.no/trygver/2003/javazone-jaoo/MVC_pattern.pdf) The model-view-controller (mvc)—its past and present
[19] van Rossum G 1995 Python tutorial *Tech. Rep. CS-R9526 Centrum voor Wiskunde en Informatica (CWI)* (Amsterdam: CWI)
[20] TIOBE 2019 TIOBE Index for July 2019 (https://tiobe.com/tiobe-index/)
[21] Millman K J and Aivazis M 2011 *Computing in Science Engineering* **13** 9–12
[22] Oliphant T E 2006 *A Guide to NumPy* vol 1 (USA: Trelgol Publishing)
[23] Jones E *et al* 2001 SciPy: Open source scientific tools for Python [Online; accessed 31 July 2019] (http://scipy.org/)
[24] Limited R C (https://riverbankcomputing.com)
[25] Qt Company (https://qt.io/)
[26] Schober A 2019 SimplePlot (https://github.com/AlexanderSchober/simpleplot_qt)
[27] Campagnola L 2019 PyQtGraph (www.pyqtgraph.org)
[28] Kindervater J *et al* 2019 in Review