

Data reduction of time-of-flight data taken on BL4A (MagRef)—a Jupyter notebook approach

The option consists of four modules (Jupyter notebooks).

- Module 1: ***BL4A\_get\_single\_event\_tutorial.ipynb*** or ***BL4A\_get\_single\_event\_batch.ipynb***. Module 1 uses Mantid to convert nexus data into NumPy binary files. Once transferred to a person's laptop, Mantid is never used again, nor is the analysis.sns.gov machine.
- Module 2: ***BL4A\_reduction.ipynb*** bins the time-of-flight data, removes instrumental background, normalizes the scattered data for the wavelength dependence of the incident beam spectrum and corrects for polarization and flipping efficiencies. The code takes into account that the as-measured direct beam is flipped relative to the as-measured specularly reflected beam.
- Module 3: ***Qz\_binner\_for\_pnr.ipynb*** Qz-bins the data to produce  $R(Q_z)$ . The code can be used for data taken with unpolarized neutron beams, polarized neutron beams with and without analysis, and for magnetic fields greater than 2 Tesla, will account for the change of energy of the neutron beam occurring during spin-flip reflection.
- Module 4: ***scale\_and\_splice.ipynb*** will scale different runs so there is approximately a seamless transition from one histogram to the next and allows user to exclude beginning and end regions of each histogram. The output is written as separate files for each histogram and as one concatenated and Qz-sorted file in tab delimited text format. These files are suitable for analysis using GenX. A recommendation is to first fit the concatenated file, then replace this file with the separate histograms to allow the computer to best-fit scale factors and scattering angle offset errors.

Steps performed to obtain reflectivity data:

- Module 1: Get your event mode BL4A data (on to your own laptop)
  - Log on to jupyter.sns.gov
  - Change directory to notebooks/REF\_M/
  - Make sure you are running a Mantid compatible kernel (I use the Python 3 Mantid kernel)
  - In first cell of *BL4A\_get\_single\_event\_batch.ipynb* type in the run numbers having the data to acquire (many files in a single directory can be specified)
  - Execute *BL4A\_get\_single\_event\_batch.ipynb*
  - Run all cells and wait. Work done by routine:
    - Removes proton flash.
    - Writes the event mode and (some) slow log data to a NumPy archive.
  - Open a terminal window on your laptop.
    - Change directory to where you want the data to go on your laptop.
    - `sftp analysis.sns.gov`
    - Change directory to target on the analysis site from the sftp shell
    - `mget *events*`
  - Terminate the terminal window and jupyter.sns.gov notebook.
  - *Issue (1): The MagH process variables are not being recorded to the log file.*

- *Issue (2): How to alert code to use Bruker or MagH process fields for B?*
- *Issue (3): Only a small selection of slow log data are written, and those that are written are assumed not to change with time.*
- Module 2: Prepare data for time-binning
  - This and the following steps are performed on your personal computer and do not require Mantid! Use a Python 3 kernel.
  - Enter USER input data in top block of ***BL4A\_reduction.ipynb***.
    - Contains the run number to analyze and its corresponding spectrum run number (as comma delimited pair).
    - Provides option to specify number of time of flight bins
    - Provides option for varying degree of sophistication for option to subtract background (if possible use the simplest option)
  - Execute ***BL4A\_reduction.ipynb*** notebook. Work done by routine:
    - Load data and the corresponding spectrum
    - Bin the data
    - Subtract background (detector face subtraction, two options available)
    - Normalize to proton charge
    - Normalize to TOF variation of spectrum (with inversion)
    - Correct for wavelength variation of polarization and flipping efficiencies (requires access to ***Efficiencies.npz***, which contains these numbers)
    - Write the results to a NumPy archive.
- Module 3: Qz-bin the results
  - Enter USER input data in 2<sup>nd</sup> block of ***Qz\_binner\_for\_pnr.ipynb***.
    - Choice of file from text events or from ***BL4A\_reduction.ipynb***
    - Non-spin-flip specular peak integration width as multiple of rms width
    - Size of Qz-bins relative to Qz—a constant
    - Choice for smallest B to impose Zeeman field splitting of neutron energy
    - *Issue (4): MagH process variable not being written to nexus file, so presently the user must set the value of B in Tesla.*
  - Auto-detect number of spin states recorded and along with value of B and Bc (=2T) decide whether or not to impose Zeeman field splitting algorithm (if B>Bc)
  - Qz-bin NSF and SF (if present) scattering.
  - Write results to GenX compatible format (a simple 4 column text format)
- Module 4: Option to scale and splice multiple files
  - Enter run files to process, i.e., scale and splice, in the first user block.
  - Execute ***scale\_and\_splice.ipynb*** notebook. Work done by routine:
    - Will plot original data
    - Will identify Qz-overlap regions between the runs and estimate a scale factor based on a weighted averaging algorithm
    - Enables the user to focus on particular overlap regions, decide how many Q-points to exclude from beginning and tail of the runs
    - Reapply the scaling algorithm

- Identify negative reflectivities, replace them with the smallest non-zero positive reflectivity and set the error bar to 1 (this will ensure these values, if there are any, do not influence fitting)
- Scale the maximum of the reflectivity to 1
- Apply all scale factors to the errors as well
- Write the scale files individually ('\_red' files) and collectively as large Qz-sorted files (look for long run title). These are tab delimited text files that are suitable for model fitting using GenX.

Advantages of the Jupyter notebook approach to data reduction on BL4A:

1. Transparency to user (code is right there in front of the user)
2. Mantid-free data treatment (runs on the user's laptop)
3. Different levels of sophistication for background fitting and subtraction, will not project background along detector-y into detector-x coordinate
4. True spectrum inversion normalization accounts for variation of spectrum across the specular beam.
5. Corrections for polarization and flipping efficiencies
6. Constant Qz-binner accounts for Zeeman splitting of the spin flip channels
7. Errors propagated through every step
8. Tab delimited text files of the specular reflectivity

An example data set consisting of three histograms (data acquired at three different angles) is included in the distribution. The data are courtesy of Dr. Ho Nyung Lee (ORNL). The example data files are too big for github but are available on zenodo.

Please cite our work.

**M.R. Fitzsimmons**, *Code to Qz-bin polarized neutron reflectivity data accounting for Zeeman energy splitting of spin-flip reflection* (2020) <https://zenodo.org/record/3828984#.XyMIQUI7IZo>  
<https://doi.org/10.5281/zenodo.3828984>

**M.R. Fitzsimmons**, *Code to map Xray reflectivity data from real space to reciprocal space* (2020)  
<https://zenodo.org/record/3967628#.XyMHd0l7IZo> <https://doi.org/10.5281/zenodo.3967628>

T.R. Charlton and **M.R. Fitzsimmons**, *Code to obtain single event data from Spallation Neutron Source Beamline-4A (MagRef)* (2020) <https://zenodo.org/record/3967680#.XyMUDUI7nBU>  
<https://doi.org/10.5281/zenodo.3967680>

**M.R. Fitzsimmons** and T.R. Charlton, *Tools to perform data reduction of time-of-flight data taken on BL4A (MagRef)—a Jupyter notebook approach* (2020)  
<https://doi.org/10.5281/zenodo.3968838>

M.R. Fitzsimmons  
 31.07.20