VESUVIO Data Reduction and Analysis in Mantid

Samuel Jackson
ISIS Facility
Rutherford Appleton Laboratory
samuel.jackson@stfc.ac.uk

August 31, 2014

Contents

1. Abstract

VESUVIO is a neutron spectrometer at the ISIS pulsed neutron source operating at high energies in the 1 eV range and uses the experimental technique known as neutron Compton scattering to measure the momentum distributions of condensed matter systems [1]. Mantid [2] is a data analysis application for neutron and muon scattering data used by multiple facilities across the world. Recently, extensive work has been carried out to integrate the bespoke data reduction and analysis routines written for VESUVIO with the Mantid framework. While the programs described in this document are designed to replicate the functionality of the Fortran and Genie routines already in use, most of them have been written from scratch and are not based on the original code base. This document outlines the current progress of development regarding what has already been implemented and what is remains to be finished.

2. Introduction

2.1. VESUVIO Overview

VESUVIO is a deep inelastic neutron scattering spectrometer situated on the S2 beamline at ISIS with an operational energy range of 5-150 eV. The instrument utilises a indirect geometry design and consists of 64 Yttrium Aluminium Perovskite γ-ray forward scattering detectors in 8 separate banks and 132 ^{6}Li doped glass scintillator back scattering detectors split into 3 banks. The scattered neutron beam impinges onto the analyzer foil which provides the energy analysis by means of resonance absorption at a given resonance energy. For this purpose usually gold foils are employed which have a large energy resonance at 4.9 eV $\pm$ 0.15 eV. With this experimental arrangement, data collection necessarily involves foil cycling in both forward and backscattering geometries in order to obtain TOF data with and without energy selection, followed by a subtraction of these two data sets to obtain NCS spectra.

In this context, it is worth emphasizing that the foil-out data set in every VESUVIO experiment is tantamount to a simultaneous ND measurement of the sample under investigation. On VESUVIO, two different final-energy-selection schemes are currently used, hereafter referred to as resonance-detector (RD) and resonance-filter (RF) configurations. The RD configuration employs the foil to define the energy of the scattered neutrons via resonant neutron absorption, and a γ-ray detector to probe the prompt cascade following the subsequent (n,γ) reaction [4]. At present, VESUVIO uses RD (RF) in forward (backscattering) geometries, respectively.

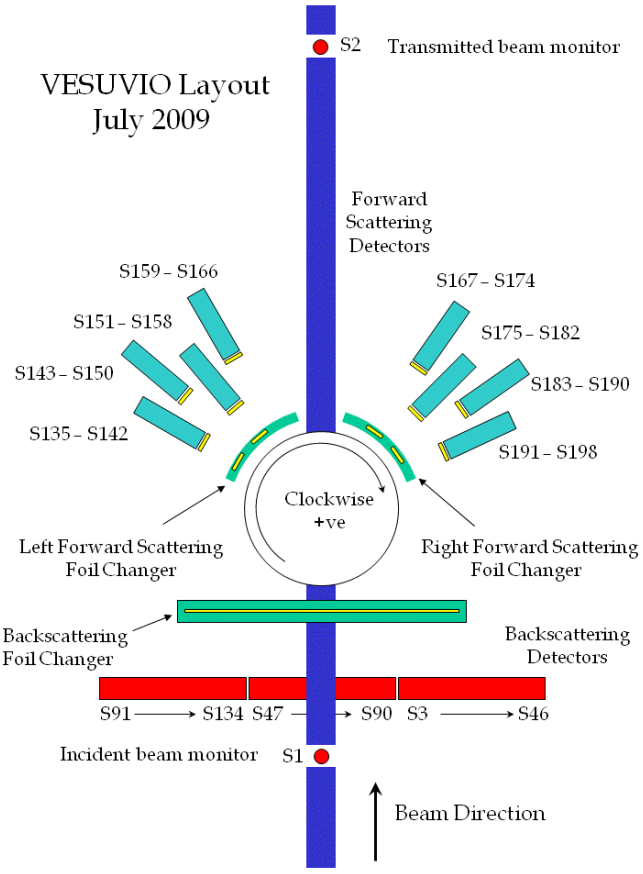

Figure 2.1: *Schematic diagram of the VESUVIO instrument. Positions of the gold foils used for each of the differencing methods are shown in yellow.*

For the RF configuration, the VESUVIO backscattering bank is equipped with Li-doped glass detectors only sensitive to neutrons [5] and the TOF spectra are obtained by taking the difference between foil-in and foil-out data, referred to as single difference (SD). To improve final-energy resolution, double-difference (DD) techniques have been implemented with three measurements, namely with no filter, a filter of

thickness d1 and neutron absorption A1(E1) and a filter of the same material of thickness d2 and absorption A2(E1). The energy resolution function is well approximated by a Voigt line shape where the Lorentzian contribution is largely removed by the DD technique.

VESUVIO is primarily designed to measure the atomic momentum distribution of atoms in condensed matter systems. Data analysis on VESUVIO relies on the accuracy of two approximations, namely the incoherent (INCA) and impulse approximation (IA). INCA implies that for incident neutron wavelengths much less than the d-spacing of the sample, atoms will scatter incoherently and the measured intensity is therefore the total sum of intensities for individual atoms in the sample [6]. IA effectively treats the scattering as single atom 'billiard ball' scattering, with conservation of momentum and kinetic energy of and a neutron a single target atom, allowing the momentum distribution of a mass to be measured using the theory developed in Ref. [1].

NCS data contains a number of spectra each consisting of a series of peaks. Each of these peaks corresponds to a single atomic mass in the sample being studied. The positions of the peaks in the spectrum are determined by the mass of the sample. The amplitude of each of the peaks is determined by an atom's mass and scattering cross section. Finally, the width of the peaks is determined by the momentum distribution of the mass. The goal of data analysis of VESUVIO data is to determine the shape and intensity of the peaks, thereby determining the momentum distributions of the atoms in the sample. [8].

2.2. Introduction to the NCS data analysis module

The Mantid development team in combination with VESUVIO instrument scientists have created a data analysis package to help aid with the analysis of VESUVIO data. This is still a work in progress and under heavy development. Currently it is in the form of a separate python module called ncs.py (NCS: Neutron Compton Scattering) which provides a collection of helper functions and classes for data analysis which builds upon the existing Mantid framework. This module is publicly available in the Mantid Github scripts repository in the *development >inelastic* folder [9]. The simplest way of including the ncs python module in Mantid is to place the files *ncs.py* and *ncs_plotting.py* in the *<Mantid Install >/scripts* folder. This will allow the ncs module to be imported directly into the Mantid script window.

3. Viewing Data in Time-of-Flight

The original data analysis package [8] used two separate commands to process the raw time-of-flight data, one for the front scattering detectors and one for the back scattering. In the Mantid implementation this has been replaced with a single algorithm called LoadVesuvio which performs all of the processing for raw files from the instrument. This includes options for handling each of the different foil positions and difference techniques available with VESUVIO [4, 7] and handles the summing multiple runs. It also contains a flag for summing each spectrum in the desired range into a single spectrum. The results of this loading operation are output as a single workspace in Mantid with units in time-of-flight which can be plotted using both Mantid's in built visualisation tools and custom plotting commands described in Ref. 10. Like all inelastic instruments at ISIS, VESUVIO uses time of flight measurements to determine the momentum and energy transfer to the neutron in each scattering event. The neutron counting chain assigns a time of flight to the arrival of each neutron at the detector.

$$t = \frac{L_0}{v_0} + \frac{L_1}{v_1} + t_0 \tag{3.1}$$

L0 is the path length from source to sample, L1 that from sample to detector, v0 is the velocity of the incident neutron and v1 that of the scattered neutron. t0 is an offset from the true value of due to delays in the electronic counting chain. On VESUVIO the values of L0, L1, v1, t0 and the scattering angle can be characterised by a mean value, with an associated distribution about the mean. Given these mean values and the value of t assigned to the neutron, the incident velocity v0 of each neutron detected and

hence the momentum and energy transfer in the scattering process can be determined from eq. 3.1 above. The distributions of these parameters about their mean values determine the resolution function of the instrument. The main task instrument calibration procedure is to determine these distributions and store them in the instrument parameter (IP) file.

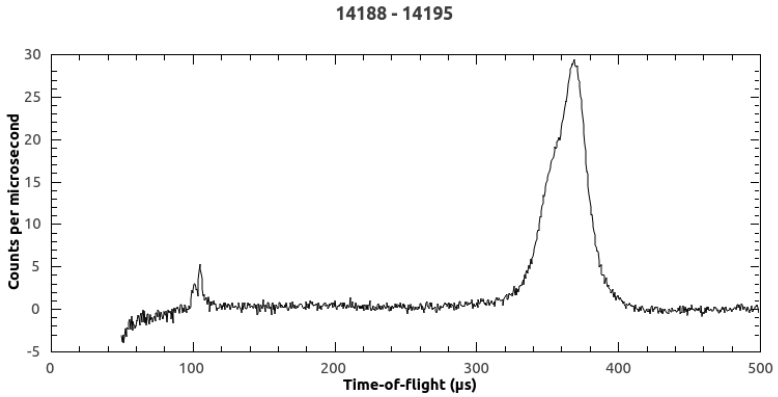

Figure 3.1: *Plot of time-of-flight data from the sum runs 14188-14195 and the sum of detectors 40-134.*

Optionally, an instrument parameter file can be supplied to the loading routine. This file contains a set of calibrated instrument parameters for each of the detectors and can correct each of the default parameters used and attach the $t0$ parameter (the time delay offset)[3] to the instrument attached to the workspace. This parameter file is usually generated by the instrument scientist using a set of reference runs using the calibration program outlined in section 8 and does not need to be regenerated for every analysis session.

Once VESUVIO data has been loaded into a Mantid workspace it can be used with the various tools in the ncs data analysis module. Typically the first step after loading the raw data in time-of-flight is to crop the workspace to a sensible range for data analysis. This can be done by using the Mantid CropWorkspace algorithm. The typical time-of-flight range used is 50.0 - 562.0 μs.

```
import ncs
runs = "14188-14195"
spectra = "134"
diff_type="SingleDifference" # Allowed values=Single,Double,Thick
ip_file = "IP0004_10.par"

raw_ws = LoadVesuvio(Filename=runs, SpectrumList=spectra,
                     Mode=diff_type,InstrumentParFile=ip_file)
raw_ws = CropWorkspace(raw_ws,XMin=50.0,XMax=562.0)
```

Listing 1: *Example script showing how to load data and crop VESUVIO data using the Mantid python API.*

The ncs.py module also provides a preprocessing function for further preparing the data for analysis. This method provides options for masking data points in the raw workspace given an error threshold as well as an option to smooth the workspace using the SmoothData algorithm. This requires a *FitOptions* object to be supplied (see section 5).

4. Multiple Scattering and Gamma Background Corrections

The time-of-flight data loaded using the methods described in section 3 need to be corrected to account for the effects of multiple scattering [10] and gamma background [11]. Currently, the multiple scattering corrections for VESUVIO in Mantid are under development and will be available in future releases. When the secondary foil absorbs neutrons it also emits -rays, which can be registered by the YAP detectors. The residual background is non-zero because the intensity of these is different in the two positions occupied

by the secondary foil. With the current arrangement of foils and detectors on VESUVIO the residual background is at the 5% level compared to the signal.

Corrections for the gamma background are implemented using the Mantid algorithm framework as an algorithm called *CalculateGammaBackground*. This takes a single workspace in time-of-flight, and a fit function describing the mass spectrum of the input data and a list of workspace indices to include as part of the correction. The fit function for the mass spectrum is typically one which has been created as part of the fit routines described in section 5. This algorithm results in two workspaces: one that contains the calculated background and a copy of the input time-of-flight workspace with the background subtracted. This function loosely corresponds to the *bcorr* command in the old data analysis package.

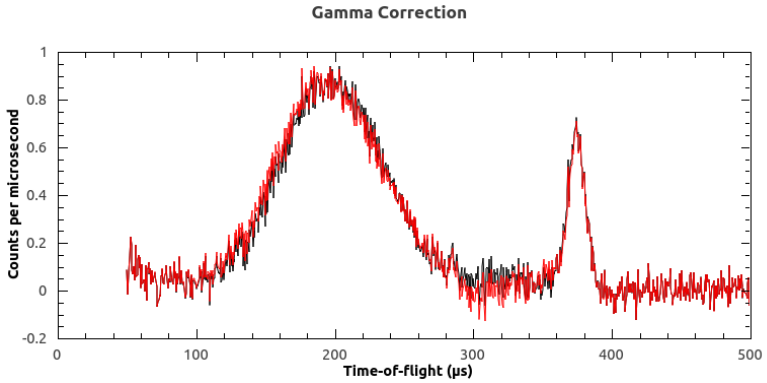

Figure 4.1: *Plot of a Zirconium Hydride (ZrH_2) sample. Black is the uncorrected sample, red shows the same sample after gamma correction.*

There are helper functions for this implemented in ncs.py. The gamma background can be computed from a workspace simply by using the *gamma_correct* function and passing the raw workspace to correct, the required fitting options, and the parameter workspace produced from a fit (see section 5). Listing 2 shows the code to perform a gamma correction on a workspace that has already been fitted.

```
ncs.preprocess(raw_ws, fit_options)
reduced_chi_square, params_ws = ncs.run_fit(raw_ws, fit_options)
ncs.display_fit_output(reduced_chi_square, params_ws, fit_options)

background, corrected = ncs.gamma_correct(raw_ws, fit_options, params_ws)
```

Listing 2: *Example script for performing a gamma correction to the ZrH_2 sample using ncs.py.*

5. Fitting

The majority of the work to integrate VESUVIO into Mantid has been concerned with the development of the fitting procedures required to measure the neutron Compton profile following the theory described in Ref. [1]. Development in this area has focussed on two major additions to the Mantid framework. First was the creation a new suite of fit functions which could accurately describe the results of a neutron Compton scattering experiment. The second was the creation of a collection of supporting data analysis functions which allow the user to easily set up a fit given the appropriate parameters for the sample and are available in the ncs.py module. These will eventually be folded into a GUI for VESUVIO (see section 9).

There are two major fit functions used in the analysis of VESUVIO data. The *GaussianComptonProfile* defines a function for fitting the simpler Gaussian approximation to mass peaks. The *GramChalierComptonProfile* is for the more complex fitting case where the atoms in the sample are affected by anisotropy and anharmonicity. The theory behind both of these functions are described in detail in Refs. [1, 12]. Both functions are implemented as standard Mantid fit functions and are therefore directly accessible through the fit wizard.

The fitting routines provided as part of ncs.py are radically different from the originals described in Ref. [8]. Currently, in order to run a fit a *FitOptions* object must be created from ncs.py and set up with the appropriate options for the sample being fitted. This includes a set of parameters for each mass describing it's atomic weight, the function it should be fitted with and any additional options specific to the model used. Listing 3 shows an example of how to set up the parameters of a fit using the ncs.py *FitOptions* class. Table A.1 lists all of the current fitting options available in the *FitOptions* class, while table A.2 describes each of the options for defining the masses to be fitted.

5.1. Fitting the data to obtain sample compositions and kinetic energies

The simplest mode of analysis is to assume that each of the peaks in momentum space is of Gaussian shape [7]. Using the amplitudes and widths obtained for the fit the composition of the sample and the atomic kinetic energies may be determined. Listing 3 shows a basic example of how to use the *FitOptions* class to set up fit for the aluminium and zirconium masses in the sample.

Fitting the data shown in listing 1 with the parameters defined in listing 3 produces three workspaces. One containing the fit of the composite function and individual mass profiles to the raw data, a table workspace of parameters for the fit, and a table workspace for the normalised covariance matrix of the fit. These workspaces are in fact just the standard output from the Mantid Fit algorithm. A plot of the fit using the Gaussian approximation function to two mass peaks for aluminium and zirconium is shown in 5.1. Comparing this to the original example in Ref. [8] shows a good correlation between implementations.

The fitted amplitudes and widths of the sample can be examined by looking at the parameters table produced by the fit. In the original VMS data analysis package a routine called *ke* was used to calculate the kinetic energy in meV and K and the fundamental frequency in meV and cm^{-1} [8]. There is currently no direct implementation of this program in Mantid.

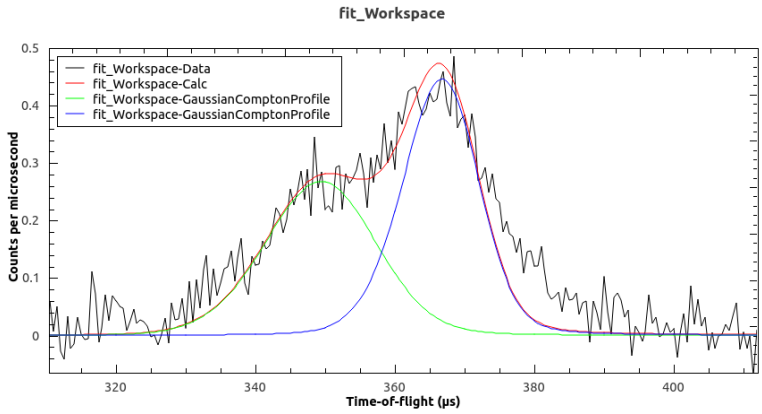

Figure 5.1: *Plot of the fitted time-of-flight data from the sum runs 14188-14195 at the scattering angle of 163 degrees for a sample of ZrH_2. The black line shows the original data, the red line shows the fit to the data is in red, and the individual contributing peaks are in green (aluminium) and blue (zirconium)*

```
fit_options = ncs.FitOptions()
fit_options.workspace_index = 0
fit_options.bad_data_error = 1e6

mass1 = {'value':27.0,  'widths':[12,14.4, 16], 'function':'Gaussian', }
mass2 = {'value':91, 'widths':[24, 26.6, 28], 'function':'Gaussian'}
fit_options.masses = [mass1, mass2]

ncs.preprocess(raw_ws, fit_options)
reduced_chi_square, params_ws = ncs.run_fit(raw_ws, fit_options)
ncs.display_fit_output(reduced_chi_square, params_ws,fit_options)
```

Listing 3: *Example script for setting up a fit to the ZrH_2 sample using ncs.py. The mass of aluminium and zirconium are fitted using the Gaussian fit function. In this example, the widths of each of the masses has been fixed using "widths" attribute*

The momentum distribution of atoms has a strictly Gaussian shape only when bound by isotropic harmonic forces. The shape of the momentum distribution $n(p)$ [7], is of interest as this contains information about the anisotropy of the binding and anharmonic effects [12]. In this case the data can be fitted using the *GramCharlier* function.

Setting up such a fit is very similar to the procedure described for the Gaussian approximation. Each mass being fitted using the expansion requires some additional parameters describing the hermite coefficients to be used in the fit and what the Final State Effects (FSE) coefficient should be tied too. See appendix A for more details on the available options. Listing 4 shows an example of setting up such a fit using the data for the ZrH_2 sample.

```
fit_options = ncs.FitOptions()
fit_options.workspace_index = 0
fit_options.bad_data_error = 1e6

mass1 = {'value':1.0079, 'widths':[2,5,7], 'function':'GramCharlier',
         'hermite_coeffs':[1,0,0],'k_free':False, 'sears_flag':1}
mass2 = {'value':27.0,  'widths':14.4, 'function':'Gaussian', }
mass3 = {'value':91, 'widths':26.6, 'function':'Gaussian'}
fit_options.masses = [mass1, mass2, mass3]

ncs.preprocess(raw_ws, fit_options)
reduced_chi_square, params_ws = ncs.run_fit(raw_ws, fit_options)
ncs.display_fit_output(reduced_chi_square, params_ws,fit_options)
```

Listing 4: *Example script for setting up a fit to the ZrH_2 sample using ncs.py. In this example, the widths of each of the masses except hydrogen has been fixed using "widths" attribute*

5.2. Fitting to determine momentum distributions

In order to fit the data to determine the momentum distribution the raw data needs to first be normalised by the area of the peak. This removes the effects of different detector efficiencies and solid angles. The Mantid algorithm *NormaliseByPeakArea* has been created to handle this step. This algorithm effectively directly replaces the implementation of the *isofile* command in the original data analysis package [8]. This algorithm will:

1. Transform the fitted data to y-space using the algorithm *ConvertToYSpace* for a single mass.
2. Fit the peak in y-space.
3. Divide the time of flight data by the area of the fitted peak in momentum space.

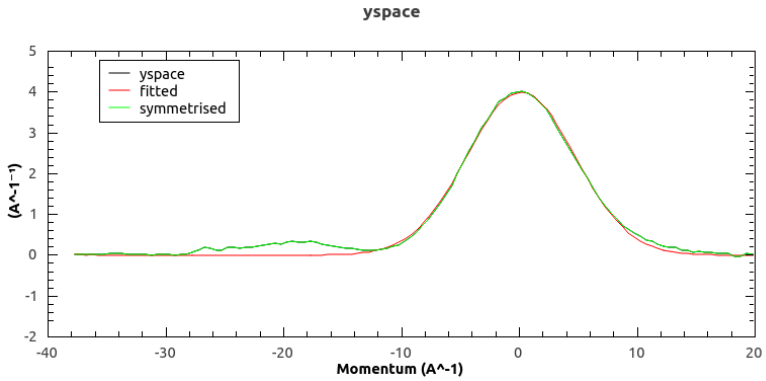

Figure 5.2: *Plot of the time-of-flight data normalised and converted to y-space as the black line (behind the green). The red line shows the fit to the peak and the green line shows the symmetrised workspace which is in good agreement with y-space data.*

The *NormaliseByPeakArea* algorithm produces four workspaces:

- **OutputWorkspace -** This workspace is the time-of-flight workspace supplied as input to the algorithm normalised by the area of the fitted peak.
- **YSpaceDataWorkspace -** This is the time-of-flight workspace converted to y-space using the *ConvertToYSpace* algorithm.
- **FittedWorkspace -** The values of the fit to the data.
- **SymmetrisedWorkspace -** The time-of-flight workspace converted to y-space and symmetrised about zero. As the momentum distribution should be symmetric about the origin, this workspace should match the *YSpaceDataWorkspace*.

```
1 normalised, yspace, fitted, symmetrised = \
2         NormaliseByPeakArea(InputWorkspace=raw_ws, Mass=1.0079,Sum=True)
```

Listing 5: *Example python code showing how to run the NormaliseByPeakArea algorithm.*

The final step in analysis is to simultaneously fit data across all detectors using the normalised workspace created as part of the previous section while keeping the widths of each peak tied. This roughly emulates the corresponding section described in Ref. [8] as the *isofileu* program. This global fitting option is still currently under development.

6. Diffraction

Diffraction on VESUVIO is not yet fully implemented within Mantid. As the reduction of diffraction data is fairly trivial in comparison to the other requirements of VESUVIO data analysis, this can be handled by the existing IndirectDiffractionReduction routine that is used by other indirect geometry instruments. Preliminary trials with the existing implementation for other indirect instruments have shown that this should simply be a matter of adding the appropriate parameters for monitor thickness, attenuation, and area to VESUVIO's instrument parameter file.

7. Resolution Calculation

Calculation of the resolution of a particular mass in a sample can be performed using the *calculate_resolution* function in ncs.py. This uses the same method to calculate the resolution that is used in the fit function described in section 5. The output of running this function is a workspace with a single spectrum for

the calculated resolution. Figure 7.1 shows a plot of the resolution for hydrogen and listing 6 shows the python code that generated the workspace shown in the plot.

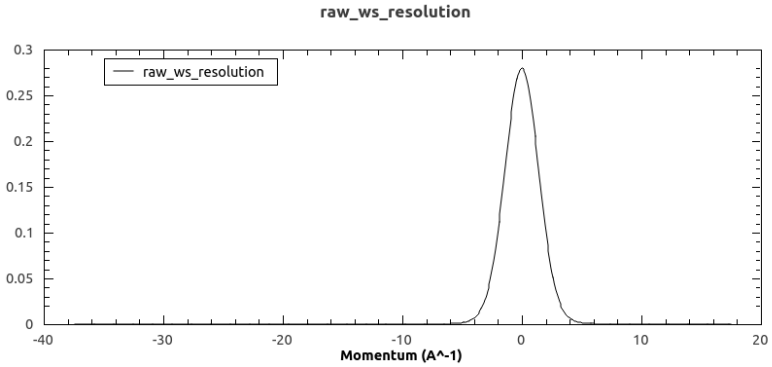

Figure 7.1: *Plot of the resolution workspace generated from listing 6.*

```
import ncs

runs = "14188-14195"
spectra = "135-198"
diff_type="SingleDifference" # Allowed values=Single,Double,Thick
ip_file = "IP0004_10.par"

raw_ws = LoadVesuvio(Filename=runs, SpectrumList=spectra,
                     Mode=diff_type,InstrumentParFile=ip_file)
raw_ws = CropWorkspace(raw_ws,XMin=50.0,XMax=562.0)

mass = 1.0079
raw_ws = ConvertToYSpace(raw_ws, mass)
ncs.calculate_resolution(raw_ws, mass)
```

Listing 6: *Example script showing how to calculate the resolution for a particular mass.*

8. Calibration of Instrument Parameters

The calibration routines for VESUVIO have been implemented as two Mantid algorithms following the procedures described in Ref. [3]. The first algorithm, called *EVSCalibrationFit*, is used to fit sample data in order to accurately obtain the values of the parameters for the instrument. A second Mantid algorithm built on top of the first is used to set up and run the fitting does the actual calculation of the instrument parameters called *EVSCalibrationAnalysis*. These two algorithms should only be run by instrument scientists and a calibration parameter file does not need to be recreated for every data analysis session.

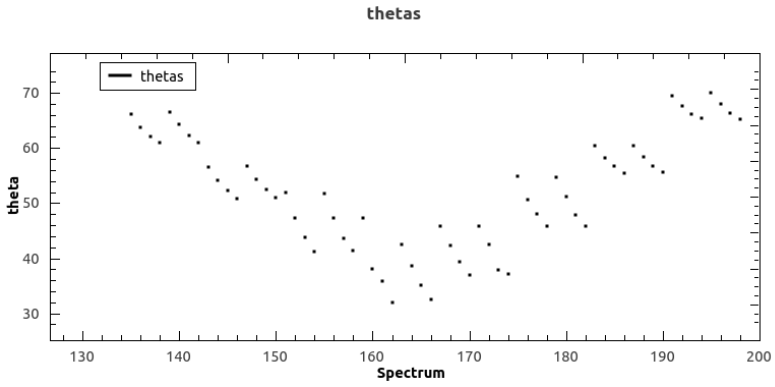

Figure 8.1: *Plot showing the calculated scattering angle from a Pb sample.*

The calibration fit program can either fit using a list of incident energies to calculate the expected centre point of a recoil peak in time-of-flight data or by taking a list of d-spacings and calculating the position of Bragg peaks. The algorithm also requires a file containing a set of reference parameters with which to calculate the expected peak positions. This is analogous to the procedures described in Ref. [3] where hand measured parameters were used as starting values and the fitting program used these to incrementally converge on the true value of the instrument parameters.

The analysis program makes use of the fit program to fit all of the parameters in the order defined by the original VESUVIO calibration paper [3], starting with the incident flight path and time delay using well defined uranium sample runs, then computing the values of the final energy and hence the final flight path and finally the scattering angles for all detectors.

9. GUI

VESUVIO does not currently have any GUI support within Mantid. Focus on development has been to get the underlying reduction and analysis routines working before focusing on usability. The current plan is for VESUVIO to have a completely new GUI listed under the indirect geometry section of Mantid. The current design would be to have a series of three tabs on the interface which would each deal with a separate part of reduction and analysis.

- **Loading:** This tab would broadly handle the procedures described in section 3 and would provide a user interface for using the LoadVesuvio algorithm to get raw data into Mantid as well as providing plotting functions for examining the captured time-of-flight data.
- **Corrections:** The corrections tab would provide a user interface for calculating both multiple scattering and gamma background corrections and applying them to data loaded in the first tab.
- **Fitting:** The fitting interface is the most complex in the series. This will provide support for setting up a fit using the same procedures used in the ncs.py module, but in a more user friendly manner than the current implementation achieves. This would provide support for both the Gaussian and Gram-Charlier functions as well as the ability to set the intensity constraints and Hermite polynomial expansion coefficients.

This is simply a loose plan based on previous discussions. The number and function of the tabs on the interface may change as development progresses. A proper design cannot be produced until the underlying framework is finalised. For example, there is a currently goal to support fitting a sample directly in y-space. It may be possible to integrate this with the fitting tab mentioned above, or depending on requirements, may be split to a fourth tab. Global fitting may well also require yet another separate interface.

10. Visualisation

VESUVIO analysis requires a few custom plotting commands to display data in ways which are not currently supported by the general Mantid framework. The final tab in the series would provide a user interface to allow the user to examine their data more easily with some helper functions which exist in the ncs_plotting.py module. This is provides a single plot function with options to plot data by spectra, scattering angles, or detector banks.

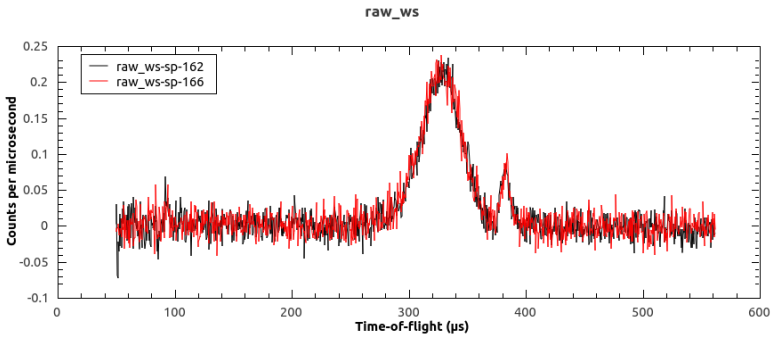

Figure 10.1: *Plot of some raw time-of-flight data using the ncs_plotting module using the script in listing 7. Checking the spectrum numbers in the key shows that the scattering angles of 33° and 32° have been plotted.*

Appendix B lists all of the currently supported plotting commands. This replicates the functionality for the various plotting routines that the old data analysis package had. The listing 7 shows a basic example of plotting some spectra from a raw workspace by angle using the ncs_plotting module. Figure 10.1 shows the resulting plot.

```
1  import ncs_plotting
2
3  runs = "14188-14195"
4  spectra = "3-195"
5  diff_type="SingleDifference" # Allowed values=Single,Double,Thick
6  ip_file = "IP0004_10.par"
7
8  raw_ws = LoadVesuvio(Filename=runs, SpectrumList=spectra,
9                       Mode=diff_type,InstrumentParFile=ip_file)
10 raw_ws = CropWorkspace(raw_ws,XMin=50.0,XMax=562.0)
11
12 ncs_plotting.plot(raw_ws, angles=(30,35))
```

Listing 7: *Example python code showing how to plot spectra in a workspace within the scattering range of 30-35° using the ncs_plotting module.*

Acknowledgments

The author would like to thank M. Gigg, A. Seel, for their knowledgeable assistance in producing this report and for aiding with the parts of the implementation the author had a hand in, and M. Gigg and R. Tolchenov who wrote the majority of the implementation presented in this document. Many thanks to M. Krzystyniak for his contributions and corrections, particularly in regard to VESUVIO theory. I would also like to thank S. Mukhopadhyay for her contributions to the report.

References

[1] J Mayers and G Reiter. *The VESUVIO electron volt neutron spectrometer*. Measurement Science and Technology, 23(4):045902, 2012.

[2] Mantid Project. *Mantid: Manipulation and Analysis Toolkit for Instrument Data*. 2013. doi: 10.5286/SOFTWARE/MANTID. URL `http://dx.doi.org/10.5286/SOFTWARE/MANTID`.

[3] J Mayers and MA Adams. *Calibration of an electron volt neutron spectrometer*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 625(1):47–56, 2011.

[4] EM Schooneveld, J Mayers, NJ Rhodes, A Pietropaolo, C Andreani, R Senesi, G Gorini, E Perelli-Cippo, and M Tardocchi. *Foil cycling technique for the VESUVIO spectrometer operating in the resonance detector configuration*. Review of scientific instruments, 77(9):095103, 2006.

[5] PA Seeger, AD Taylor, and RM Brugger. *Double-difference method to improve the resolution of an eV neutron spectrometer*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 240(1):98–114, 1985.

[6] F. Fernandez-Alonso and D. L. Price. *Neutron Scattering Fundermentals*. Academic Press, 1st edition, 2013.

[7] J Mayers, J Tomkinson, T Abdul-Redah, WG Stirling, C Andreani, R Senesi, M Nardone, D Colognesi, and E Degiorgi. *VESUVIO - the double difference inverse geometry spectrometer at ISIS*. Physica B: Condensed Matter, 350(1):E659–E662, 2004.

[8] J. Mayers. *User guide to VESUVIO data analysis*. Technical Report RAL-TR-2011-003, Rutherford Appleton Laboratory UK, 2010.

[9] *Mantid Github Scripts Repository*. `https://github.com/mantidproject/scripts/tree/master/development/inelastic`. Accessed: 2014-08-11.

[10] J Mayers, AL Fielding, and R Senesi. *Multiple scattering in deep inelastic neutron scattering: Monte Carlo simulations and experiments at the ISIS eVS inverse geometry spectrometer*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 481(1):454–463, 2002.

[11] J Mayers. *Calculation of background effects on the VESUVIO eV neutron spectrometer*. Measurement Science and Technology, 22(1):015903, 2011.

[12] C Andreani, D Colognesi, J Mayers, GF Reiter, and R Senesi. *Measurement of momentum distribution of lightatoms and molecules in condensed matter systems using inelastic neutron scattering*. Advances in physics, 54(5):377–469, 2005.

Appendices

A. Fit Options

The following two tables list all of the current options that may be supplied to the *FitOptions* object created using ncs.py. The first table shows the general parameters for the fit, while the second table lists the options used to define a single mass used as part of a fit.

Name	Description
smooth_points	Number of data points to use in the SmoothData algorithm in the pre-processing function. If not set the data will not be smoothed.
bad_data_error	If set, the data will be masked if it falls outside of this error tolerance.
background_function	The type of background to use in fitting. By default this is set to use the Polynomial background function.
background_order	The order of the background to use.
masses	A dictionary of parameters for each mass supplied to the fit. This should include the atomic weight, function to fit (either *Gaussian* or *GramCharlier*) any any other parameters required by the function.
constraints	Constraints on the intensity of each mass peak.
workspace_index	The index in the workspace to fit to.
output_prefix	String which is prefixed to output workspaces created from the fitting.
global_fit	Whether or not to perform a multi-dataset (global) fitting of the data.

Table A.1: *Table listing the different fitting options available in the FitOptions class*

Name	Description
value	The atomic weight of the mass in amu
widths	The width of the peak. If this is set to a single number the width will be fixed to that value. If it is set to an iterable (tuple or list) of three values the middle value is the starting value and the first and last values are the constraints on the parameters' fit.
function	The function to use to fit this mass. Options are currently either *Gaussian* or *GramCharlier*.
hermite_coeffs	Only applicable when using *GramCharlier*. This is a list of coefficients for each mass being fitted. If set to one the term is included within the fit. If set to zero it is excluded from the fit
k_free	Only applicable when using *GramCharlier*. If this flag is set to true the FSE ("Final State Effects") coefficient is not tied to a value. If false it is set to a value dependant on the *sears_flag*.
sears_flag	Only applicable when using *GramCharlier*. If the *k_free* flag is set to false, this flag will control what the FSE coefficient gets tied too. If set to one it is tied to $width \times \sqrt{\frac{2}{12}}$ which is harmonic limit of the magnitude of the FSEs

Table A.2: *Table listing the options for defining a mass to be fitted.*

B. Plotting Options

The following table summarises the plotting options available in the *plot* routine in the ncs_plotting.py module.

Name	Description
spectra	Plot a single spectrum or, given a list of indices, plot any number of spectra.
angles	Plot by scattering angle given a list/tuple of size two specifying the minimum and maximum angles in the range.
bank	Plot a single bank or, given a list of bank numbers, plot any number of banks with the inclusive range of 1-8.
errors	Create the plot with error bars shown. Default is false.
sum	Sum all spectra falling within the range defined by the angle/spectra/bank options. This is applied independently to each workspace is there is more than one. Default is false.
fig	If supplied the a new window will not be created, but the new plot will be attached to the existing figure referenced by *fig*
clrfig	If supplied the given window will be clear before the new plot is drawn.

Table B.1: *Table listing the options that can be used in conjunction with the plot function.*