VESUVIO Data Analysis Goes MANTID

S. Jackson\textsuperscript{1,2}, M. Krzystyniak\textsuperscript{1,3}, A.G. Seel \textsuperscript{1}, M. Gigg\textsuperscript{1,4}, S.E. Richards\textsuperscript{1,3}, and F. Fernandez-Alonso\textsuperscript{1,5}

\textsuperscript{1} ISIS Facility, Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire OX11 0QX, United Kingdom
\textsuperscript{2} Department of Computer Science, University of Aberystwyth, Penglais, Aberystwyth, SY23 3DB, United Kingdom
\textsuperscript{3} School of Science and Technology, Nottingham Trent University, Clifton Campus, Nottingham NG11 8NS, United Kingdom
\textsuperscript{4} Tessella Ltd., Abingdon, Oxfordshire, UK
\textsuperscript{5} Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom

Abstract. This paper describes ongoing efforts to implement the reduction and analysis of neutron Compton scattering data within the MANTID framework. 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.

1. Introduction

Neutron Compton scattering (NCS) may be regarded as a mass spectroscopic technique in which each atomic mass contributes to the overall time-of-flight (TOF) spectrum in the form of a Doppler-broadened recoil peak. The width of such recoil peak in the energy domain can be related to the momentum distribution of the target nucleus prior to the scattering effect. NCS has been successfully applied in a wide variety of condensed matter systems and molecules. This includes fluid and solid hydrogen [1, 2, 3]; water [4, 5], proton glasses [6], biological molecules [7, 8], hydrogen storage materials [9, 10], helium ($^3$He and $^4$He) [11, 12, 13, 14, 15, 16], noble gases in solid and fluid phases [17, 18], graphite [19], metal hydrides [20], etc.

NCS data-reduction schemes can be conveniently subdivided into two groups: (i) those that aim solely at the determination of scattered intensities [21, 22], and (ii) those that serve the purpose of the determination of kinetic energies and momentum distributions of individual nuclei in a mass-selective manner [23, 24, 25, 26, 27, 28]. The use of existing NCS data treatment schemes usually requires remote access to a variety of different computing environments with routines compiled for different operational systems and hardware infrastructures. These requirements substantially increase the complexity and time for the data analysis and reduction stage. The first step towards decreasing this complexity was made by extending the capabilities of the first generation of NCS data reduction routines by translating them into a set of MATLAB routines within a common NCS toolbox where computation, visualization, and programming can be completed within the same environment. Nevertheless, both FORTRAN and MATLAB versions of the NCS software have been developed and maintained in a closed environment...
resulting with suboptimal code development. Therefore, consolidation and translation of the NCS data reduction routines into the MANTID framework has become a prerequisite both for further code and technique development. MANTID, with extensive use of PYTHON and C++, uses a common interface which is seamless between the facility, beam line and home institution when performing the final analysis and fitting of the data [29]. Furthermore, this approach increases transparency between the developer and user community. This has been the main motivation behind the work presented here. In this paper, we briefly describe VESUVIO data-reduction and analysis as implemented in Fortran, MATLAB, and MANTID, and use the example of zirconium hydride as a test bed to demonstrate their equivalence. Examples of the NCS data-reduction routines using the MANTID graphical user interface and PYTHON scripts are presented. In the final section additional functionalities are discussed in the context of their potential of providing new insights in the analysis of NCS data.

2. Neutron Compton scattering and the impulse approximation

In NCS, the energy and momentum transfers imparted by the neutron to the target nucleus are so high that the scattering process can be treated within the (IA) [30, 31, 23]. The IA effectively treats NCS as billiard-ball scattering with conservation of kinetic energy and momentum of the neutron and a given (struck) nucleus. Under the IA, the scattering function, $S_M(q, \omega)$ for a given nucleus of mass $M$ reduces to a single peak centered at the recoil energy $E_r = \hbar^2 Q^2 / 2M$. In this limit, $S_M(q, \omega)$ has the form [23]:

$$S_M(q, \omega) = \int n_M(p)\delta\left(\omega + \frac{p^2}{2M} - \frac{(p + q)^2}{2M}\right)$$

(1)

where $n_M(p)$ is the nuclear radial momentum distribution for mass $M$. Consequently we can write $S_M(q, \omega) = \frac{M}{\hbar^2 q} J(y, \hat{q})$, where:

$$y = p \cdot \hat{q} = \frac{M}{\hbar^2 q} (\omega - \omega_r) = \frac{M}{\hbar^2 q} \left(\omega - \frac{\hbar^2 q^2}{2M}\right)$$

(2)

where $\hat{q}$ is a unit vector in the direction of the momentum transfer. Hereinafter, momentum transfer will be given in units of $A^{-1}$, energy transfer $\omega$ in meV, and atomic mass in amu. With this choice of units, Plank’s constant is given by $h = 2.0446$ (meV amu)$^{1/2}$ Å.

$J(y, \hat{q})$ is the so called neutron Compton profile [31, 30] and is formally the result of a Radon transformation of the radial momentum distribution $n_M(p)$. The relationship given by Eq. (2) is formally known as ‘y-scaling’ or the ‘West-Scaling’ [32, 31, 23]. It implies that $q$ and $\omega$ are no longer independent variables and that for samples with no preferred orientation any scan in $q - \omega$ space crossing the recoil line, defined by the relation $\omega_r = \frac{\hbar^2 q^2}{2M}$, gives a measurement of the neutron Compton profile $J(y, \hat{q})$. For a harmonically bound isotropic system where the momentum distribution $n_M(p)$ depends only on the magnitude of the radial momentum, $p = |p|$, and the neutron Compton profile $J(y, \hat{q})$ in the IA limit, hereinafter denoted as $J_{IA}(y)$, can be written as a normalized Gaussian [33, 34, 35, 36, 23] of the form

$$J_{IA}(y) = \frac{1}{\sqrt{2\pi} \sigma_p^2} \exp\left(\frac{-y^2}{2\sigma_p^2}\right)$$

(3)

where $\sigma_p$ is the standard deviation. Equation is exactly correct when the atoms are bound by isotropic harmonic forces. In real data, the anisotropy and anharmonicity of the local binding potential both give a non-Gaussian $J_{IA}(y)$. Equation (3) is generally good when small atomic displacement from the mean position sample the harmonic region. The current resolution
and count rates on VESUVIO make the measurement of any small deviations from Gaussian behaviour for $M > \sim 20$ impractical.

Even when the IA is not satisfied, $y$ scaling can still be an appropriate starting point. For finite values of $q$, corrections to the IA known as 'final-state effects' (FSEs) are usually expressed using $y$ scaling [31, 36, 23]. To account for FSEs the method of Sears is routinely incorporated into standard NCS data treatments [31, 36, 23] by expressing the measured neutron Compton profile $J(y)$ to first order as

$$J(y) = J_{IA}(y) + J_{FSE}(y) = J_{IA}(y) - \frac{M \langle \nabla^2 V \rangle}{36h^2 q} \frac{d^3}{dy^3} J_{IA}(y) + ...$$

(4)

where $J_{IA}(y)$ is the IA result. $\langle \nabla^2 V \rangle$ is the mean value of the Laplacian of the potential energy of the nucleus expressed in meVÅ$^{-2}$ [cf. Refs. 19, 36, 23]. In an NCS experiment performed on an isotropic sample within the IA limit, a given NCS profile $J_A(y)$ can be related to a spherically averaged three-dimensional nuclear momentum distribution (NMD) $n_{IA}(p)$ by the following expression [23]

$$n_{IA}(p) = -\frac{1}{2\pi y} \frac{\partial J_{IA}(y)}{\partial y} \bigg|_{y=p}$$

(5)

where the radial NMD $n_{IA}(p)$ can be represented as

$$n_{IA}(p) = \exp \left( \frac{-p^2}{2\sigma^2} \right) \sum_n c_n (-1)^n L_n^{1/2} \left( \frac{p^2}{2\sigma^2} \right)$$

(6)

where $L_n^{1/2}$ denotes a generalised Laguerre polynomial. Equation (6) can be used for the reconstruction of $n_{IA}(p)$ from the measured $J(y)$ after subtraction of the FSE contribution $J_{IA}(y) = J(y) - J_{FSE}$ where

$$J_{IA}(y) = \exp \left( \frac{-y^2}{2\sigma^2} \right) \sum_n \frac{c_n}{2^{2n}n!} H_{2n} \left( \frac{y}{\sigma\sqrt{2}} \right)$$

(7)

where $H_{2n}$ denotes a Hermite polynomial. In the above expression, the coefficients $c_n$ are identical to the expansion coefficients for the radial momentum distribution, which is tantamount to an inversion procedure between $J_{IA}(y)$ and $n_{IA}(p)$. The expansions are in fact over complete: $\sigma$ can be arbitrary and the series in Eq. (7) will still converge satisfactorily to an arbitrary function [27]. However, if the first term is omitted in the fitting (that is $c_1$ is fixed at zero), $\sigma$ necessarily determines the kinetic energy, independently of any other terms in the series.

3. Count rate on an inverse-geometry instrument

On an inverse geometry instrument like VESUVIO at ISIS [27], the time-of-flight (TOF), $t$, of each detected neutron is determined by

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

(8)

where $L_0$ is the source-sample distance, $L_1$ is the sample-detector distance, $v_0 = \sqrt{2E_0/m}$ is the incident neutron velocity, and $E_0$ the incident neutron energy, $v_1 = \sqrt{2E_1/m}$, the velocity corresponding to detected (and thus fixed) neutron energy $E_1$, and $m$ is the neutron mass. The incident energy $E_0$, can be calculated from the measured TOF, $t$ (Eq. 8), given $L_0$, $L_1$, and $E_1$. Hence the energy transfer
\[ \omega = E_0 - E_1 \] (9)

and the momentum transfer

\[ q = \sqrt{2m} \left( E_0 + E_1 - 2\cos\theta \sqrt{E_0 E_1} \right)^{1/2} \] (10)

can both be determined from \( t \).

The count rate as a function of \( t \) on an inverse-geometry neutron spectrometer can be written as [27]

\[ C(E_0) \, dE_0 = 2 \left( \frac{2}{m} \right)^{1/2} \frac{\left[ E_0 \right]^{3/2}}{L_0} I(E_0) D(E_1) N \frac{d^2\sigma}{dE_1 d\Omega} \, d\Omega \] (11)

where \( I(E_0) \, dE_0 \) is the number of incident neutrons per second per unit area with energies between \( E_0 \) and \( E_1 + dE_0 \). \( D(E_1) \) is the probability that a neutron energy \( E_1 \) is detected, \( \frac{d^2\sigma}{dE_1 d\Omega} \) is the double differential cross-section, \( N \) is the number of atoms in the beam, and \( d\Omega \) is the detector solid angle.

The spectra are, however, recorded and represented as a function of \( t \). The Jacobian \( \frac{dE_0}{dt} \) is given by the following expression [36]:

\[ \frac{dE_0}{dt} = \frac{2}{L_0} \sqrt{\frac{2}{m} E_0^{3/2}} \] (12)

Using the Jacobian \( \frac{dE_0}{dt} \) the count rate \( C_\theta(t) \) in a given TOF channel \( t \) becomes [36]:

\[ C_\theta(t) = \left[ \frac{E_0 I(E_0)}{q} \right] (t) \times A' M I_M J_M [y_M(t)] \] (13)

where \( A' \) is another mass-independent experimental constant.

If atoms of different masses \( M \) are present in the sample, the total count rate can be generalised to [36]:

\[ C_\theta(t) = A' \left[ \frac{E_0 I(E_0)}{q} \right] (t) \times \sum_{n=1}^{M} I_n M_n J_n [y_n(t)] \] (14)

where both the experimental constant \( A' \) and the factor \( \left[ \frac{E_0 I(E_0)}{q} \right] (t) \) are mass independent and can be written in front of the sum over \( M \). The integrated peak intensity \( I_M \) for a given mass \( M \) is given by \( I_M = A N_M \sigma_M \), where \( \sigma_M = 4\pi b_M^2 \) is the total bound scattering cross section.

4. Instrument resolution

In a real experimental situation the measured count rate \( C_\theta(t) \) for every TOF, \( t \), is an average over the possible values of all geometrical parameters characterising neutron trajectories and scattering angles as well as the final neutron energy weighted by its probability of occurrence. In case of NCS data reduction the most frequently adopted formalism to account for the effects of instrument resolution is to replace the multiple integrals over distributions of the above-mentioned parameters that would have been required to be calculated for each data point in a TOF spectrum for a given mass by convolution with a mass-dependent resolution function. In the framework of the so-called ‘convolution approximation’ (CA) it is assumed that a TOF spectrum for a given mass is described by a convolution of a mass-dependent resolution function of the instrument \( R[y_M(t)] \) and the neutron Compton profile of the scatterer \( J[y_M(t)] \) [36]. Typically, the resolution function \( R[y_M(t)] \) is assumed to be described by a Voigt line shape as
it is the case for the gold foil used on VESUVIO [36]. The convolution is performed in y-space for each mass $M$ and then the sum over contributions from different masses is performed to account for the total count rate recorded in a given TOF spectrum. Thus, for a total of $N$ different masses present in the sample, the total count-rate at a fixed scattering angle $\theta$, $C_\theta(t)$ is given by [36]:

$$C_\theta(t) = A' \left[ \frac{E_0 I(E_0)}{q} \right] (t) \times \sum_{n=1}^{m} I_n M_n j_n [y_n (t)] \otimes R_n [y_n (t)]$$

where $A'$ is a mass-independent experimental constant and the mass-independent factor $\left[ \frac{E_0 I(E_0)}{q} \right] (t)$ depends on the incident neutron spectrum, $I(E_0(t))$ the initial neutron energy, $E_0(t)$, and the momentum transfer $q(t)$, all functions of the TOF, $t$ [36]. The difference between the instrument resolution function implemented in [27] and the instrument resolution function implemented in the NCS reduction routine used in the MANTID environment is a convolution in y-space rather than $t$ [27].

FSE contributions to both Gaussian and non-Gaussian $J_M$ are accounted for by including terms proportional to $1/qH_3$. The resulting expression to fit the entire TOF spectrum is of the following form [37]:

$$C_\theta(t) = A' \left[ \frac{E_0 I(E_0)}{q} \right] (t) \times \left[ J_H (x_H) \otimes R_H (x_H) + \sum_{M \neq H} J_M (x_M) \otimes R_M (x_M) \right]$$

where $J_H$ is the longitudinal nuclear momentum distribution for the first mass (i.e. the mass for which the recoil peak is centered at the lowest TOF value)

$$J_H (x_H) = \frac{\exp \left( -\frac{x_H^2}{2\sigma^2_H} \right) }{2\pi^{3/2}} \left( 1 + \frac{c_4}{32} H_4 (x_H) - \frac{k}{q} H_3 (x_H) \right)$$

and $J_M$ is the longitudinal nuclear momentum distribution for any other subsequent mass $M$

$$J_M (x_M) = \frac{\exp \left( -\frac{x_M^2}{2\sigma^2_M} \right) }{2\pi^{3/2}} \left( 1 - \frac{k}{q} H_3 (x_H) \right)$$

with $x_M = \frac{y_M - y_0 M}{\sqrt{2}\sigma_M^2}$, where $y_0 M$ is the shift of the position of the maximum of a nuclear momentum distribution from the center of the recoil line and $\sigma_M$ is the standard deviation of the momentum distribution for a mass $M$. The FSE expansion in the limit of a local effective harmonic potential for a given mass $M$ leads to a very useful expression of the FSE coefficient $k$ in the expansion given by $k = \sigma_M^2$ [37]. Thus, fitting of the recoil peaks in a harmonic approximation does not require any extra fitting parameter to describe FSEs.

5. VESUVIO data analysis in MANTID

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` 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 [38]. 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 PYTHON module to be imported directly into the MANTID script window.
5.1. Data loading and pre-treatment options

The original data analysis package [39] 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 [40, 36] 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 built-in visualisation tools and custom plotting commands described in Ref. [39](see Fig. 1).

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 according to Eq. (7). On VESUVIO the values of geometrical parameters defining the TOF, $L_0$, $L_1$, $v_1$, $t_0$, and the scattering angle, $\theta$, can be characterised by mean values, with associated distributions about the mean. Given these mean values and the value of TOF assigned to the neutron, the incident velocity, $v_0$, of each neutron detected and hence the momentum and energy transfer in the scattering process can be determined from Eq. (7). 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 1](image.png)

**Figure 1.** Plot of time-of-flight data from the sum runs 14188-14195 and the sum of detectors 40-134.

The parameter file is usually generated by the instrument scientist using a set of reference runs using the calibration program outlined in section 5.8 and does not need to be regenerated for every analysis session.

The data pre-treatment procedure in the MANTID NCS reduction scheme includes: definition of differencing technique for final-energy analysis (i.e., single or double differencing) [40, 41]
followed by smoothing and truncation of data points. Residual $\gamma$-background correction, multiple-scattering correction and baseline fitting complete the data pre-treatment procedure.

5.2. Definition of differencing technique

$\gamma$-ray signals are detected on VESUVIO neutron detectors at forward-scattering angles [42]. These are made to function both as neutron detectors and energy selectors. In forward scattering, a foil-cycling (FC) technique is employed whereby two foils of the same resonance material are used, one stationary analyzer foil and one cycling foil that alternates in and out of the scattered neutron beam [40]. The energy $E_1$ of detected neutrons is by a large neutron absorption resonance peak at $E_1 = 4.9$ eV. By placing a primary gold foil on the YAP detector surface, the foil absorbs neutrons of energy $E_1 \approx 4.9$ eV and emits a $\gamma$-ray cascade which can be detected, thereby determining both the TOF and the final energy of the neutron. These measurements are called 'foil-out' measurements because neutrons do not have to pass through the secondary gold foil. The $\gamma$-ray signal produced from the 'foil-out' measurements sits on a background, which may be due to $\gamma$-ray rays produced when neutrons initially scattered by the sample are then absorbed by the surrounding shielding. By positioning the secondary foil between the sample and the detector, the signal produced from these scattered neutrons are significantly reduced as neutrons of energy $E_1 \approx 4.9$ eV have to first pass through the secondary foil (or filter) to reach the primary foil. Through the use of this secondary gold foil, the energy resolution is also improved and the background almost eliminated. The VESUVIO 'raw' data is the difference between these two counts ('foil-out' minus 'foil-in') [42, 43].

In backscattering, gold foils are placed in front of $^6$Li-doped neutron detectors. In this case, a double-difference (DD) technique [41] is used to remove the background and sharpen the energy resolution. This method is based on a linear combination of two measurements recorded with filter foils of the same resonance material of different thickness. Both the FC and DD methods give a value of 4.897eV for the mean value of $E_1$ [44]. Detailed descriptions of the energy resolution functions can be found in Refs [41, 40, 45, 46].

On VESUVIO, 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 is commensurate with the size of a data vector used in an FFT-based convolution routine used fitting, i.e., 50.0 - 562.0 $\mu$s (see Table 1).

Table 1. Example script showing how to load data and crop VESUVIO data using the MANTID PYTHON API.

```python
1 import ncs
2 runs = "14188-14195"
3 spectra = "134"
4 diff_type="SingleDifference" # Allowed values=Single, Double, Thick
5 ip_file = "IP0004_10.par"
6 raw_ws = LoadVesuvio(Filename=runs, SpectrumList = spectra,
7 Mode=diff_type, InstrumentParFile = ip_file)
8 raw_ws = CropWorkspace(raw_ws, XMin=50.0, XMax=562.0)
```

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 5.5).
5.3. γ-background corrections in forward scattering

The γ-background correction utilizing the differencing techniques described above significantly reduces the γ-ray signals arising from the absorption of nuclei from the boron shielding on VESUVIO. However, the data from the YAP detectors in forward scattering still needs to be corrected for potential, sample-dependent (residual) γ-backgrounds because the intensity of the γ-background from the secondary foils in the instrument is dependent upon their position, which differs in the foil-in and foil-out measurements. This may result in spurious broad peaks in the baseline of the spectrum. Therefore the residual γ-background is calculated from the measured data \[42\] as:

\[ \eta C_{out} - C_{in} = CA^2 + (B_{out} - B_{in}) \] (19)

where \( C_{out} \) is the ‘foil-out’ count-rate from the primary gold foil, \( C_{in} \) is the ‘foil-in’ count-rate from the secondary gold foil, \( A(E_1) \) is the analyzer-foil neutron absorption coefficient at final energy \( E_1 \), and \( \eta \) is the probability that the subsequent γ-ray cascade is registered by the YAP detector. \( B_{out} \) is the γ-background with the secondary foil in the ‘out’ position, and \( B_{in} \) is the background with the secondary foil in the ‘in’ position.

Corrections for the gamma background are implemented using the MANTID algorithm framework as an algorithm called \texttt{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 5.5. 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 \texttt{bcorr} command in the old data analysis package.

![Gamma Correction](image)

\textbf{Figure 2.} Plot of a Zirconium Hydride (ZrH\textsubscript{2}) sample. Black is the uncorrected sample, red shows the same sample after gamma correction.

There are helper functions for this implemented in \texttt{nces.py}. The gamma background can be computed from a workspace simply by using the \texttt{gamma correct} function and passing the raw workspace to correct, the required fitting options, and the parameter workspace produced from
a fit (see section 5.5). Table 2 shows the code to perform a gamma correction on a workspace
that has already been fitted. The result of the code application is shown in Figure 2.

Table 2. Example script for performing a gamma correction to the ZrH$_2$ sample using ncs.py.

| Line | Code |
|------|------|
| 1    | ncs.preprocess(raw_ws, fit_options) |
| 2    | reduced_chi_square, params_ws = ncs.run_fit(raw_ws, fit_options) |
| 3    | ncs.display_fit_output(reduced_chi_square, params_ws, fit_options) |
| 4    | background, corrected = ncs.gamma_correct(raw_ws, fit_options, params_ws) |

5.4. Multiple-scattering correction
Multiple scattering (MS) occurs because there is always a finite probability that neutrons will
scatter more than once in the sample before being detected. MS must be subtracted from
the TOF spectra before accurate results for the momentum distribution can be obtained. MS
corrections are calculated using the Monte-Carlo method described in reference [47]. Currently,
the multiple scattering corrections for VESUVIO in MANTID are under development and will
be available in future releases.

5.5. Fitting Routine
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 Section 3. 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 5.9).

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 GramCharlierComptonProfile is for the more complex fitting case described
by Gram-Charlier expansion [5, 6]. Both functions are implemented as standard MANTID fit
functions and are therefore directly accessible through the fit wizard. 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. Table 3 shows an example of how to set up the parameters of a fit using the ncs.py
FitOptions class. Table 4 lists all of the current fitting options available in the FitOptions class,
while Table 5 describes each of the options for defining the masses to be fitted.

The simplest mode of analysis is to assume that each of the peaks in momentum space is of
Gaussian shape [36]. Using the amplitudes and widths obtained for the fit the composition of
the sample and the atomic kinetic energies may be determined. Table 6 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 Figure 2 with the parameters defined in Table 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
Table 3. Example script for setting up a fit to the ZrH2 sample using \textit{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.

```python
1 fit_options = ncs.FitOptions()
2 fit_options.workspace_index = 0
3 fit_options.bad_data_error = 1e6
4 mass1 = {'value':27.0, 'widths':[12.0,14.4,16.0], 'function':\'Gaussian\'}
5 mass2 = {'value':91, 'widths':[24.0,26.6,28.0], 'function':\'Gaussian\'}
6 fit_options.masses = [mass1, mass2]
7 ncs.preprocess(raw_ws, fit_options)
8 reduced_chisquare, params_ws = ncs.run_fit(raw_ws, fit_options)
9 ncs.display_fit_output(reduced_chisquare, params_ws,fit_options)
```

Table 4. Table listing the different fitting options available in the \textit{FitOptions} class.

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

\textit{MantidFit} algorithm. A plot of the fit using the Gaussian approximation function to two mass peaks for aluminium and zirconium is shown in Figure 3. Comparing this to the original example in Ref. [44] 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 \textit{ke} was used to calculate the kinetic energy in meV and K and the fundamental frequency in meV and cm$^{-1}$ [39]. There is currently no direct implementation of this program in MANTID.

The momentum distribution of atoms has a strictly Gaussian shape only when bound by isotropic harmonic forces. The shape of the nuclear momentum distribution is of interest as this contains information about the anisotropy of the binding and anharmonic effects [23]. In this case the data can be fitted using the Gram Charlier expansion [28]. Setting up such a fit is
Table 5. Table listing the options for defining a mass to be fitted.

| 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 dependent 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 $\sigma \frac{7}{12}$ which is harmonic limit of the     |
|            | magnitude of the FSEs.                                                     |

Figure 3. 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.
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. Table 6 shows an example of setting up such a fit using the data for the ZrH$_2$ sample.

**Table 6.** 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.

```python
1 fit_options = ncs.FitOptions()
2 fit_optionsworkspace_index = 0
3 fit_optionsbad_data_error = 1e6
4 mass1 = {'value':1.0079, 'widths':[2,5,7], 'function':'GramCharlier',
5 'hermite_coeffs':[1,0,0], 'k_free':False, 'sears_flag':1}
6 mass2 = {'value':27.0, 'widths':14.4, 'function':'Gaussian'}
7 mass3 = {'value':91, 'widths':26.6, 'function':'Gaussian'}
8 fit_optionsmasses = [mass1, mass2, mass3]
9 ncs.preprocess(raw_ws, fit_options)
10 ncs.display_fit_output(reduced_chi_square, params_ws, fit_options)
```

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 [39]. This algorithm will:

(i) Transform the fitted data to y-space using the algorithm `ConvertToYSpace` for a single mass.

(ii) Fit the peak in y-space.

(iii) Divide the time of flight data by the area of the fitted peak in momentum space.

The `NormaliseByPeakArea` algorithm (see Table 7) 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`.

**Table 7.** Example PYTHON code showing how to run the NormaliseByPeakArea algorithm.

```python
1 normalised, yspace, fitted, symmetrised =
2 NormaliseByPeakArea(InputWorkspace=raw_ws, Mass=1.0079, Sum=True)
```
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 (see Figure 4). This roughly emulates the program isofileu described in Ref. [39]. This global fitting option is still currently under development.

![Figure 4](image)

**Figure 4.** 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.

5.6. 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.

5.7. 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 5.5. The output of running this function is a workspace with a single spectrum for the calculated resolution. Figure 5 shows a plot of the resolution for hydrogen and Table 8 shows the PYTHON code that generated the workspace shown in the plot.

5.8. Calibration of Instrument Parameters
The calibration routines for VESUVIO have been implemented as two MANTID algorithms following the procedures described originally in Ref. [44]. The first algorithm, called
**Figure 5.** Plot of the resolution workspace generated using data in Table 8.

**Table 8.** Example script showing how to calculate the resolution for a particular mass.

```python
1 import ncs
2 runs = "14188-14195"
3 spectra = "135-198"
4 diff_type="SingleDifference" # Allowed values=Single, Double, Thick
5 ip_file = "IP0004_10.par"
6 raw_ws = LoadVesuvio(Filename=runs, SpectrumList=spectra,
7   Mode=diff_type, InstrumentParFile=ip_file)
8 raw_ws = CropWorkspace(raw_ws, XMin=50.0, XMax=562.0)
9 mass = 1.0079
10 raw_ws = ConvertToYSpace(raw_ws, mass)
11 ncs.calculate_resolution(raw_ws, mass)
```

*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.

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. [7] 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 [44], 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 (see Fig. 6).

5.9. 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 5.1 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.
5.10. 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.

This replicates the functionality for the various plotting routines of the old data analysis package [39]. Table 9 shows a basic example of plotting some spectra from a raw workspace by angle using the NCS plotting module. Figure 7 shows the resulting plot.

### Table 9. Example PYTHON code showing how to plot spectra in a workspace within the scattering range of 30-35 degrees using the NCS plotting module.

```python
import ncs_plotting
runs = "14188-14195"
spectra = "3-195"
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)
ncs_plotting.plot(raw_ws, angles=(30,35))
```

![Figure 7](image_url)
6. Conclusion
MANTID, the Manipulation and Analysis Toolkit for Instrument Data, is an open source and cross platform data analysis application specialising in neutron and muon scattering data. Recent development work undertaken to improve support for the indirect geometry spectrometer VESUVIO at ISIS, described in this work, demonstrates how MANTID can contribute to the success of neutron Compton scattering at providing unique insights into nuclear quantum dynamics of light nuclei in condensed matter and Mass-selective Neutron Spectroscopy of heavier nuclei important in the studies of functional materials.

7. Acknowledgments
The authors gratefully acknowledge the UK Science & Technology Facilities Council for financial support under Contract No. PR1200068 and access to beamtime at the ISIS Facility.

[1] Celli M, Zoppi M and Mayers J 1998 Phys. Rev. B 58 242
[2] Bafite U, Celli M, Zoppi M and Mayers J 1998 Phys. Rev. B 58(2) 791
[3] Zoppi M, Celli M and Soper A K 1998 Phys. Rev. B 58 11905
[4] Pietropaolo A, Senesi R, Andreani C, Botti A, Ricci M and Bruni F 2008 Phys. Rev. Lett. 100 127802
[5] Pantalei C, Pietropaolo A, Senesi R, Imberti S, Andreani C, Mayers J, Burnham C and Reiter G 2008 Phys. Rev. Lett. 100 177801
[6] Feng Y, Ancona-Torres C, Rosenbaum T, Reiter G, Price D and Courtens E 2006 Phys. Rev. Lett. 97 145501
[7] Senesi R, Pietropaolo A, Bocedi A, Pagnotta S and Bruni F 2008 Phys. Rev. B 58 242
[8] Bafile U, Celli M, Zoppi M and Mayers J 1998 Phys. Rev. B 58 242
[9] Zoppi M, Celli M and Soper A K 1998 Phys. Rev. B 58 11905
[10] Mayers J, Burke T and Newport R 1994 Journal of Physics-Condensed Matter 6 641
[11] Celli M, Zoppi M and Mayers J 1998 Physical Review B 58 242
[12] Peck D, Fujita I, Schmidt M and Simmons R 1992 Physical Review B 86 4584
[13] Fradkin M, Zeng S X and Simmons R 1994 Physical Review B 45 9680
[14] Fielding A, Timms D N and Mayers J 1998 Europhysics Letters 44 255
[38] https://github.com/mantidproject/scripts/tree/master/development/inelastic
[39] Mayers J 2011 User guide to vesuvio data analysis programs for powders and liquids Tech. Rep. RAL-TR-2011-003 Rutherford Appleton Laboratory, Chilton, UK URL http://purl.org/net/epubs/manifestation/5989
[40] Schooneveld M, Mayers J, Rhodes N, Pietropaolo A, Andreani C, Senesi R, Gorini G, Perelli-Cippo E and Tardocchi M 2006 Review of scientific instruments 77 095103
[41] Seeger A, Taylor A D and Brugger R M 1985 Nucl. Instrum. Meth. A 240 98
[42] Mayers J 2011 Measurement Science and Technology 22 015903
[43] Mayers J and Adams M A 2011 Nucl. Instrum. Meth. A 625 47
[44] Fielding A and Mayers J 2002 Nucl. Instrum. Meth. A 480 680
[45] Andreani C, Colognesi D, Degiorgi E, Filabozzi A, Nardone M, Pace E, Pietropaolo A and Senesi R 2003 Nucl. Instrum. Meth. A 497 535
[46] Pietropaolo A, Andreani C, Filabozzi A, Pace E and Senesi R 2007 Nucl. Instrum. Meth. A 570 498
[47] Mayers J, Fielding A and Senesi R 2002 Nucl. Instrum. Meth. A 454 481