Investigating multiple scattering with McStas

James Tricker

Contents

[Introduction 2](#_Toc492633858)

[Multiple scattering algorithms currently available in Mantid 3](#_Toc492633859)

[Mayers Sample Correction 3](#_Toc492633860)

[Multiple Scattering Cylinder Absorption 4](#_Toc492633861)

[Muscat 5](#_Toc492633862)

[Comparison using a cylindrical vanadium sample 5](#_Toc492633863)

[Cylinder absorption corrections 5](#_Toc492633864)

[Multiple scattering and cylinder absorption corrections 6](#_Toc492633865)

[Simulating multiple scattering in McStas 8](#_Toc492633866)

[McStas concepts 8](#_Toc492633867)

[Initial investigation 9](#_Toc492633868)

[Histogram data 10](#_Toc492633869)

[Future development 13](#_Toc492633870)

[Carpenter out-of-plane cylinder absorption improvements 13](#_Toc492633871)

[Multiple scattering inside infinite plane slab 14](#_Toc492633872)

[McStas S(q,w) procedure 14](#_Toc492633873)

[Appendix 15](#_Toc492633874)

[A. Cylinder absorption comparison python script (Polaris 95598) 15](#_Toc492633875)

[B. Absorption and multiple scattering python script (Polaris 95598) 17](#_Toc492633876)

[C. Incoherent vanadium sample McStas instrument file 18](#_Toc492633877)

[D. McStas compilation/simulation batch script for Windows 19](#_Toc492633878)

[E. Mantid McStas loader python script 20](#_Toc492633879)

[References 22](#_Toc492633880)

# Introduction

Determination of the structure of samples in neutron diffraction relies on the assumption that neutrons enter a sample, are diffracted once and then are picked up by a detector. In any real experiment there will be some amount of absorption by the sample and a small number of neutrons may scatter more than once before reaching the detector. Although in many typical experiments multiple scattering (MS) is often negligible, these effects can occasionally add considerable artefacts to the data which must then be corrected for.

To get some idea of when multiple scattering corrections are needed let us now define the total cross section per atom as the sum of its scattering and absorption cross sections:

\sigma_t = \sigma_s + \sigma_a

If \sigma_m is the likelihood of a neutron being scattered m times then it is possible to show [1] that:

\sigma_m \sim (\frac{\sigma_s}{\sigma_t})^m

So we can see that higher order terms tend to zero at a rate determined by the ratio of the scattering cross section to total cross section. Where practical, the shape and thickness of a sample are carefully chosen to minimise as much unwanted MS as possible. This may be achieved by using a sample that is either [2]:

* Small in comparison with its mean free path.
* Strongly absorbing (the absorption cross section is much greater than the scattering cross section. Usually this means the dimensions of a sample are chosen to ensure that between 10% and 20% of incident neutrons end up being scattered) [3].

Increasing the absorption cross section is not always attainable - due to the type of material in question - or desirable, due to prohibitive intensity losses.



Figure 1 – How a general double scattering process might occur within a sample

In Figure 1, we see a neutron which travels a certain distance l_1 through a sample before the first scattering event. The second scattering occurs after a distance l_{12} has been traversed after which the neutron travels a final length l_2 before leaving the sample and being picked up by a detector.

The difficulty in correcting multiple scattering arises from the fact that in order to analytically calculate the contribution from mth-order scattering we must perform a volume integral m times (dV_1dV_2...dV_m) over the sample. Although successive terms rapidly become negligible as explained above, this kind of calculation is incredibly demanding for all but the simplest of geometries (i.e. cylindrical, planar or spherical).

In some areas, such as small angle scattering, there may be useful approximations that can be applied that are not present for the more general wide angle scattering case. Again matters may become complicated, as for example small angle scatter followed by incoherent scatter from hydrogen can be more significant in blurring sharp features than double small angle scatter.

# Multiple scattering algorithms currently available in Mantid

The two main algorithms available in Mantid are the Mayers Sample Correction (Mayers) and the Multiple Scattering Cylinder Absorption Correction (Carpenter). These algorithms both work by calculating both an absorption factor and a multiple scattering factor and then removing these from the data. Most of this section focusses on the functioning and comparison of these two algorithms. Both Mayers and Carpenter make the problem more tractable by limiting the sample geometry to that of a cylinder and only working with isotropic, elastic scatterers (this effectively restricts us to vanadium). A third less understood algorithm, Muscat also exists and is briefly mentioned.

### Mayers Sample Correction

The full background to Mayers is described by Lindley et al. in [1]. The algorithm calculates and applies corrections due to the effects of absorption (plus optionally multiple scattering) on the signal and error values for a given workspace. The following assumptions are made:

* the sample shape is a cylinder
* for multiple scattering:
  + the scattering is assumed to be elastic and isotropic
  + the ratio of successive orders of scattering are all equal to \beta(\theta, \phi, E)

The aim is to correct the number of neutrons at a given detector (N_d) to compute the corrected number of neutrons (N_c) that would have reached the detector if there was no absorption (or multiple scattering). Ignoring detector efficiency we can write:

N_d = \frac{N_c}{A_s} \cdot \frac{1}{1-\beta}

Where A_s(\theta, \phi, E) the absorption and self-shielding factor is computed by numerical integration over the sample cylinder.

The multiple scattering factor (if requested) is computed by simulating over a fixed number of second order scattering events and computing the ratio of second order and first order scattering. Since we have assumed the ratio is the same between successive orders, the \frac{1}{1-\beta} factor simply comes from taking the sum of an infinite geometric series.

The cylinder radius r combined with the inverse attenuation length \mu = \mu(E) (derived from the total scattering cross-section) gives a range of \mu r against input time-of-flight (“energy”) for the cylinder. The \mu r range is divided into a discrete number of points for each point.

A weighted least-squares fit is applied to both the set of attenuation and multiple scattering factors to allow interpolation of the correction factor from any time-of-flight value in the input range. For each time-of-flight value the factor is computed from the fit coefficients and the correction applied multiplicatively:

\begin{aligned}
y_{out} &= y_{in} * corrfact \\
e_{out} &= y_{out} * e_{in} / y_{in}
\end{aligned}

The above procedure is repeated separately for each spectrum in the workspace.

### Carpenter Multiple Scattering Cylinder Absorption

Multiple Scattering Cylinder Absorption works in a similar manner to the Mayers Sample Correction. While Mayers works in workspace units of Time-of-flight, Carpenter works in units of wavelength.

In [4] we see that the calculation of the attenuation factor F involves an integral over a sample cylinder. By expanding the integrand as a power series, we can then, for each term, factor out any dependence on scattering cross section and radius from a dimensionless integral which depends only on scattering angle. These integral terms are denoted by Z_{mn} and so the attenuation factor can be expressed as:

\frac{1}{F} = \sum_{m=0}^\infty\sum_{n=0}^\infty\frac{(-1)^{m+n}}{m!n!}(\mu R)^{m+n} Z_{mn}(\theta)

Where, once again, \mu is the inverse scattering length and elastic scattering is assumed so that we can take the incident and scattered cross sections to be equal.

The functions Z_{mn}(\theta) have already been solved and have been expressed in a basis of Chebyshev polynomials for which the coefficients c_{s}(m,n) up to m + n \leqslant 5 are tabulated and stored in an array within the code for this algorithm:

Z_{mn}(\theta) = \sum_{s=0}^\infty c_{s}(m,n)cos(s\theta)

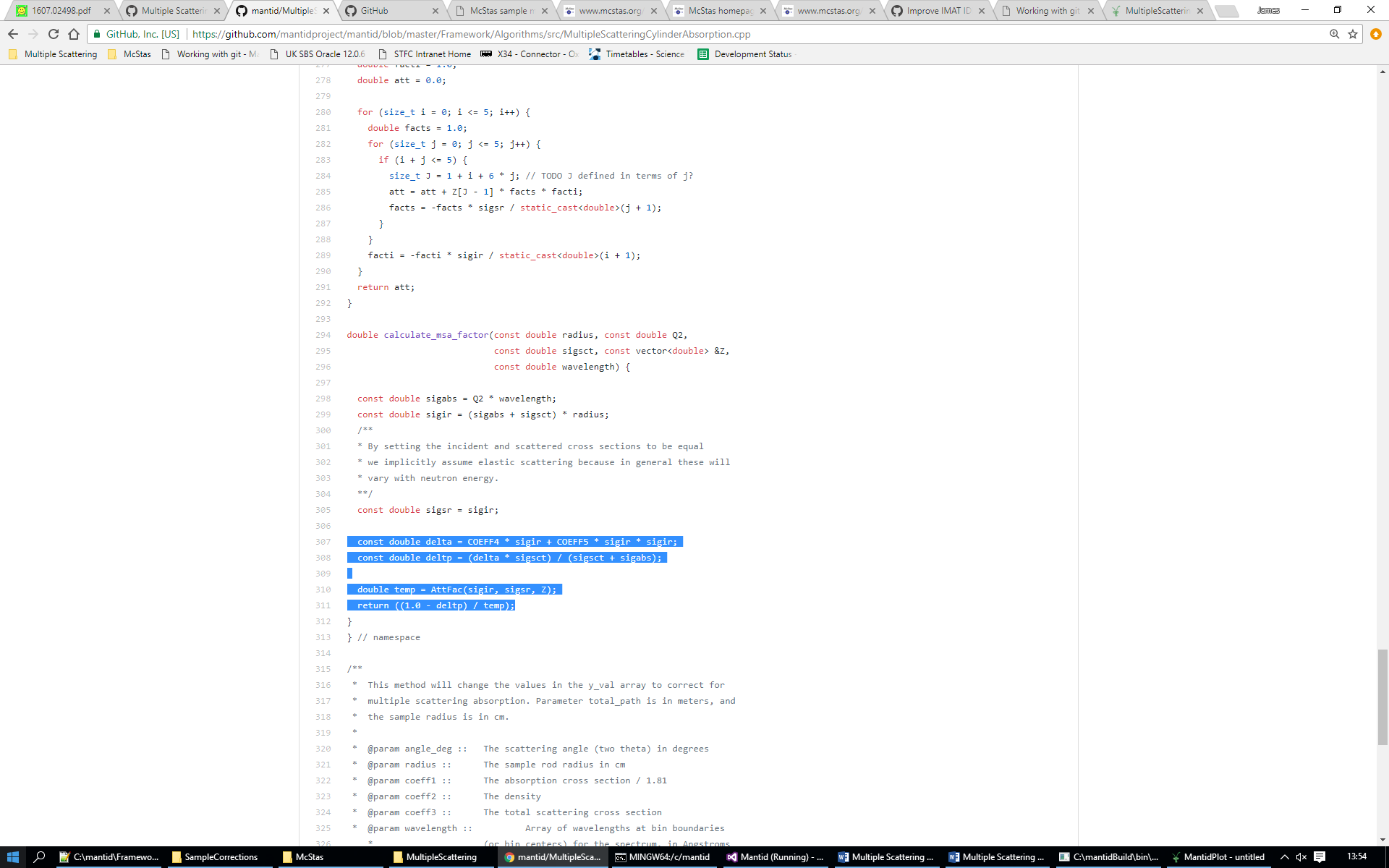
It is possible to quantify the accuracy of Carpenter for a given order of expansion [6] and it would be relatively simple to program the code to go to higher orders.

As for the multiple scattering factor, it is not clear quite how Carpenter works out the factor that accounts for multiple scattering. The only paper referenced by the algorithm which mentions multiple scattering does so only very briefly [5]:

“… let PI be the idealized single scattering probability in the absence of attenuation or multiple scattering, and let PI and PM be, respectively, the actual probabilities for single and multiple (more than single) scattering/=. Then P1/ PI is the single scattering attenuation correction factor, and (P1 + PM)/ PI is the over-all correction factor for both attenuation and multiple scattering. In writing

The over-all correction factor appears as the product of the single scattering attenuation correction factor and the factor (1 + PM/PI), which accounts for multiple scattering.”

Searching within the code for the implementation of this equation leads to the following around line 300 within the calculate\_msa\_factor function:



The constants COEFF4 and COEFF5 are defined earlier on in the code (around line 70) with no indication of precisely where or how they have been calculated or even defined. Searching in some of the literature did throw up a paper [7] where there is a method of approximating the transmission factor (above called the absorption factor) as an analytic expression in powers of \mu r up to order two. It is likely that a similar approach has been used here although for completeness it would be ideal to find some source of or method of reproducing the working that determined these coefficients in order to document it alongside the algorithm.

### Muscat

Mayers and Carpenter are both designed for elastic scattering. The algorithms MuscatData and MuscatFunction are based on the Monte Carlo program MINUS and may be used to calculate MS for flat and cylindrical geometries [13] using a general scattering function S(Q,w) (i.e. a function of Q, momentum transfer, and omega, energy exchange). The scattering function is either input directly (MuscatData) or calculated first from certain functions before being passed to the actual algorithm (MuscatFunction).

The algorithm was written a considerable time ago in FORTRAN and is run at the moment using F2Py (a FORTRAN to Python converter) and has not been successfully re implemented in Python yet due to do the confusing logic. Nonetheless Muscat provides an interesting example of how to solve multiple scattering more generally.

## Comparison using a cylindrical vanadium sample

### Cylinder absorption corrections

Mayers Sample Correction and Multiple Scattering Cylinder Absorption both work out the absorption and multiple scattering corrections as two separate factors before applying them to the input workspace. It was decided to first assess the accuracy of just the absorption factors, with the algorithm Cylinder Absorption (CA) providing a useful guideline for comparison. This analytic algorithm is well trusted and has previously been tested and verified against alternative Monte Carlo methods.

Mantid Python script used to generate the results is located in Appendix A however bear in mind that it was necessary to make a minor modification to the Carpenter code in order to disable the multiple scattering correction and isolate solely the absorption correction. The data used were the Polaris run 95588 on a cylindrical sample of vanadium. The ratios of the Mayers and Carpenter outputs against CA are shown in Figure 2. A horizontal line close to unity is what we look for to verify the accuracy of these algorithms and this is broadly what we see.

The slight deviation of the Carpenter from horizontal (shown by the red line) is hypothesised to be accounted for by the fact that Carpenter does not currently handle out of plane scattering. It would be interesting to see if extending the Carpenter to cope with this led to any improvement and a later section outlines how this may be done. Examining the Carpenter line in closer detail we see that it remains smooth over a larger interval of d-spacing than the Mayers correction shown in black (which on a finer scale is much more jagged relative to the Carpenter line and also starts to really diverge from our "true" correction above 3 Angstroms).

It is likely Mayers could be made to produce a smoother line (i.e. more precise output) if the step intervals of integration were to be made smaller. Since the absorption correction is not currently very time consuming this could probably be done without drastically increasing run time (the multiple scattering correction is by far the most time consuming step).

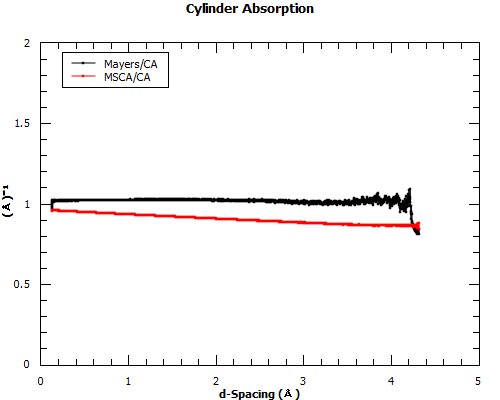


Figure 2 – Mayers in black and Carpenter in red. Both are normalised by the Carpenter

### Multiple scattering and cylinder absorption corrections

To fully compare the Mayers and Carpenter corrections the script in Appendix B was produced which is simply an adapted version of the previous script, following a similar scheme but with both absorption and multiple scattering corrections now being applied.

In Figure 3 the almost horizontal line showing the ratio of the two outputs suggests that the algorithms are in reasonably close agreement if we ignore the overall scale factor. Since the algorithms use different methods to compute their outputs this is an encouraging confirmation that they both are generally correct

To account for the slight slope seen in Figure 3 we refer back to Figure 2 and notice that it is roughly the same as the slope of the Carpenter line. It is likely that any major deviation between the algorithms in Figure 3 is solely due to the differences in the absorption correction factors[[1]](#footnote-1).

It is noteworthy that Mayers takes significantly longer than Carpenter to run (about fifteen minutes versus at most a minute). This is because the Mayers calculation of the multiple scattering correction factor involves making multiple simulations of multiple scattering events which, for Carpenter, is done using a simple two term power series expansion in \mu r as explained in the Carpenter introduction section. The fact that the results are fairly concordant (aside from the discrepancy due to the absorption corrections) is perhaps an indication that the simulation method used by Mayers to calculate MS is excessive for the majority of cases since Carpenter is able to produce similar results in a much shorter timeframe using a far simpler method.

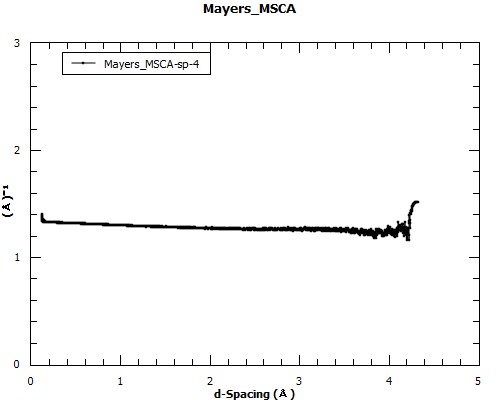


Figure 3 - the ratio of the curves (with both corrections now applied) showing an almost horizontal line

It was briefly considered that some consolidation of Mayers/Carpenter could reduce unnecessary overlap of code features however due to the differences in the two algorithms this is unlikely to be possible. A more viable improvement would be to look into replacing the cylinder absorption method within Carpenter and replace it with a call to the Cylinder Absorption algorithm. This would remove the need to improve Carpenter to handle out-of-plane scattering which is not guaranteed to actually even improve the algorithm and would likely require considerably more effort (none-the-less implementing out-of-plane corrections is still explored later in this report).

The currently suggested improvements to Mantid going forward are to create a CalculateAbsorptionAndMultipleScatteringCylinder algorithm which will then give the user the option to choose either the Carpenter or Mayers method. It would output separate workspaces with the correction factors for absorption and multiple scattering which the user would then have to separately apply to their original workspace. The current algorithms MultipleScatteringCylinderAbsorption and MayersSampleCorrection would use the new algorithm to calculate the correction and then apply it to the data as before.

# Simulating multiple scattering in McStas

## McStas concepts

The McStas [9, 10] neutron ray-tracing simulation package is a versatile tool for producing accurate simulations of neutron scattering instruments at reactors and pulsed spallation sources. McStas is extensively used for design and optimization of instruments, virtual experiments, and user training. McStas is an abbreviation for Monte Carlo Simulation of triple axis spectrometers, but allows for describing and simulating any type of neutron scattering instrument. [11]

The software works by assembling a C file from instrument information written in a C-like syntax. This code can then compiled by McStas and run to simulate neutron scattering experiments. One of the main purposes of this investigation is to see if McStas combined with Mantid can provide a workable solution for calculating multiple scattering for general samples of any geometry.

The McStas Wiki page on McStas and Mantid [12] provides a good introduction on how to enable the output of Mantid compatible data using their software. Here is an excerpt which explains the basic procedure:

“Generating McStas event data for Mantid is a bit different from running a standard McStas simulation. This is related to the way Mantid expects to read in event data. The generation of McStas event data for Mantid is a tree step procedure.

* From the McStas instrument file generate the IDF file for Mantid.
* Compile the McStas c code. Remember to link correctly to NeXus libraries.
* Run McStas simulation with the NeXus flag on to produce event data for Mantid.

Step 1 will general an xml file for Mantid with a description of the instrument geometry, which is known as the Instrument Definition File (IDF). The IDF is used for TOF conversions in Mantid. Step 2 is used to make sure you McStas simulation can produce a NeXus file as output. Step 3 is to run the McStas simulation and ask for a NeXus file as output of the simulation, as opposed to several ASCII files which is the default setup in McStas.”

When repeatedly making small changes to instrument files this procedure quickly became tedious and so a simple but valuable batch script (Appendix D) was written in Windows which expediently ran through these steps to conveniently produce the data when required.

One of the main benefits of using McStas is the ability to determine for each neutron simulated how many times it has been scattered before reaching the monitor put in place to detect it. The aim was to configure McStas to output these different neutron events into separately loadable workspaces - All, Multiple and Single – allowing a richer analysis of multiple scattering to be done in Mantid.

The basic approach for utilising McStas and Mantid integration for multiple scattering is described in [11] and goes like this:

1. Generate a scattering function S = S(Q) or S(Q, omega) from the data in question using Mantid
2. Making the assumption that multiple scattering for S does not contain multiple scattering and is good first order approximation of the true S of the sample
3. Insert this S into a McStas component and do a McStas simulation where the output of this simulation is separated into single and multiple scattered neutrons

The dataset of multiply scattered neutrons then represents our first order approximation of the multiple scattering. This procedure might then be carried out iteratively, calculating a new S(Q,w) from the single scattered events and reapplying the whole approach once again. If this procedure was successfully implemented then the convergence (or lack thereof) after several iterations would be a good initial indicator of success or failure.

This procedure appears rather similar to that used by Muscat, both work with the scattering function of the sample in order to correct the multiple scattering. It would be interesting to examine how the Muscat algorithm manages to perform the Monte Carlo simulations and determine if this part could be somehow adapted to use an engine based on McStas code which would allow for more general sample shapes and even containers to be simulated. Regardless, Muscat is still an interesting point of reference in designing an algorithm that follows a scattering function based procedure and at least some of the work may need not be repeated.

## Initial investigation

The original plan of the investigation was to follow an analogous scheme to that used in the previous comparison except this time comparing workspaces rather than algorithms. The All and Single scattering workspaces could be loaded into Mantid and then Cylinder Absorption[[2]](#footnote-2) applied to the Single workspace to produce an Ideal workspace containing no multiple scattering or absorption. Carpenter and Mayers can then each be applied to the All workspace and this should in principle be identical to the Ideal workspace.

The method of integrating McStas and Mantid intended by the McStas team and directed in [12] meant that each simulated neutron event would be recorded totalling 6 parameters logged for each event: intensity, x and y position, neutron ID, pixel ID and time of flight. The monitors within the instrument file (Appendix C) had options setting parameters that looked like this:

options ="mantid square, x limits=[-2.5 2.5] bins=50 y limits=[-2.5 2.5] bins=50, neutron pixel t limits [0.002,0.005] list all neutrons, file=all"

Where ‘file’ varied depending on what type of scattering the detector was configured to measure. When loading the data generated by this into McStas using the LoadMcStas algorithm, a workspace was generated that had an X axis giving Time-of-flight and a Y axis showing counts (similar to how the Polaris 95598 data from the previous section were presented) along with 2500 spectra corresponding to each of the virtual detectors determined by the 50 by 50 bin widths in x and y. This is a complete way of storing the data however it would be preferable if McStas were to histogram the data into detector bins “on-the-fly” rather than having to measure single neutrons events which are tallied to detectors when the data are loaded. Logging individual events seem to crash McStas very easily and put an upper limit of about 10e6 on the number of simulations than could be done in a single. Attempting to perform 10e7 simulations led to the following message in the McStas shell:

Events: "all\_list.p.x.y.n.id.t"

Monitor\_nD: nD\_Mantid\_2 cannot reallocate Vars->Mon2D\_Buffer[0] (6480000). Skipping.

Monitor\_nD: nD\_Mantid\_1 cannot reallocate Vars->Mon2D\_Buffer[0] (1760000). Skipping.

Meaning that the monitors in place to measure the multiple and single scattering events were skipped when the number of neutrons got too large.

A get around initially explored was to run the simulation twice on two different instrument files, using the same seed, one recording all the events and one recording only single scattering events. Unfortunately this method did not lead to very much improvement. Although a data file was sometimes produced for a run with 10e7 neutrons, it was unreadable in the standard HDF5 viewer due to its excessive size (containing over 10 million individual neutron events each with 6 unique measurements). In other cases the data were not even able to be output and McStas giving the error message:

Events: "single\_list.p.x.y.n.id.t"

Trace ETA 4.1 [min] % 4 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95

Save [Vanadium1]

# McStas 2.4.1 - Jun. 26, 2017: [pid 6216] Signal 11 detected SIGSEGV (Mem Error)

# Simulation: Vanadium1 (VanadiumSeed\_single.instr)

# Breakpoint: nD\_Mantid\_0 (Save) 100.00 % (100000000.0/100000000.0)

# Date: Thu Aug 31 11:45:22 2017

# Started: Thu Aug 31 11:37:33 2017

# Last I/O Error: No error

# McStas 2.4.1 - Jun. 26, 2017: Simulation stop (abort).

When loading the outputted file into Mantid the intention was that the analysis would be done in a way much similar to that done using the Polaris data. Examining the workspace, each detector contained counts only in a narrowly defined bin of time-of-flight which was unlike the real data used previously. It could be that this was due to the parameters of the incoming neutron beam and the geometry of the instrument and detector being set up in McStas in a way that was somehow unsuitable to producing data that could be interpreted in the same way as before. A more varied selection of instrument configurations would be needed to make sure this is not the case.

## Histogram data

Another possible explanation for the difference in data was that 10e6 simulations did not produce enough points to eradicate fluctuations and gaps. A practicable workaround to the problem of having a limited number of runs was to forgo instructing the McStas instrument to record detector position information and instead simply configure it to measure the flux between bins of time-of-flight. This was done by modifying the ‘options’ settings for the monitors to:

options ="mantid square, time limits [0.002,0.005], flux, file=all"

This is not how McStas was intended to be used with Mantid and as a result the IDF files ended up being poorly embedded in the data file and needed to be loaded manually, as well as the dimensions and units of the data requiring to be set manually so that Mantid could interpret them correctly. The main downside was that now there was no recorded information about where on the monitor neutrons had been detected which could be used to separate counts into individual spectra. Mantid responded to this by putting all the counts into the very first spectra corresponding to meaningless detector location which had no relevance to the position defined in the instrument file.

The lack of accurate information about the location of detected neutrons was a worrying omission however it none the less allowed some basic work to be done on the results Mantid using the script in Appendix E.

It is questionable to what extent the results shown in Figure 4 are truly meaningful since no specific detector information is recorded and so Mantid interprets all counts as belonging to Spectra 0 as if there was only a single detector. The monitor in McStas fills a definite area which could potentially be spanned by several detectors and the correction in general will depend on their positioning.

Despite these considerations the top two lines of Figure 5 confirm that both of the algorithms have actually performed changes to the workspace. The bottom two lines indeed show the result that we would hope to see if we were looking to confirm the consistency between the algorithms and the multiple scattering simulation done by McStas. The fact that we see a horizontal line close to unity in spite of the issues with detector correspondence and positioning is intriguing. One possible explanation is that the effects vary very little with detector position; in which case the actual position of the detectors is unimportant to the final correction.

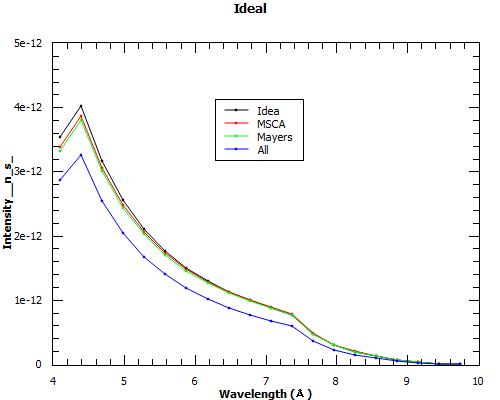


Figure 4 – plots of the three workspaces which in theory should now be identical (to within a scale factor)

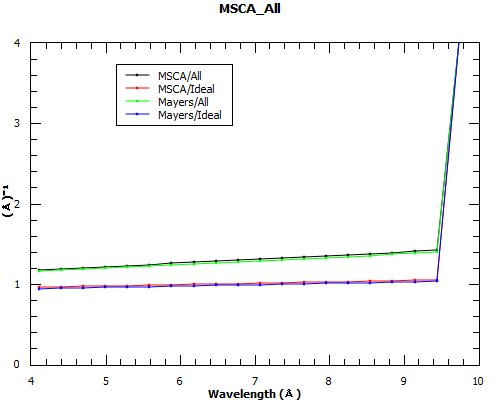


Figure 5 – the ratios of the computed results vs the idealised one and vs the All workspace

A brief and rudimentary inquiry into the suggestion that detector positon was not that important meant going back to the data from the initial McStas investigation where detector position was actually determinable. The fact that all the data were placed in a single time-of-flight bin was less of a hindrance this time since now the only item of interest was the variation of the correction between detectors. The left hand side of Figure 6 shows a column containing a set of correction factors (for Carpenter in this case) which have all been divided by the same name number, giving a set of values all very close to one.

All of the figures were consistently very close to one with seemingly no systematic pattern to what little fluctuation was present. More thought is necessary in order to truly interpret what is going on here but hopefully the investigation and scripts provided represent some preliminary use cases for what can be done when combining Mantid and McStas.

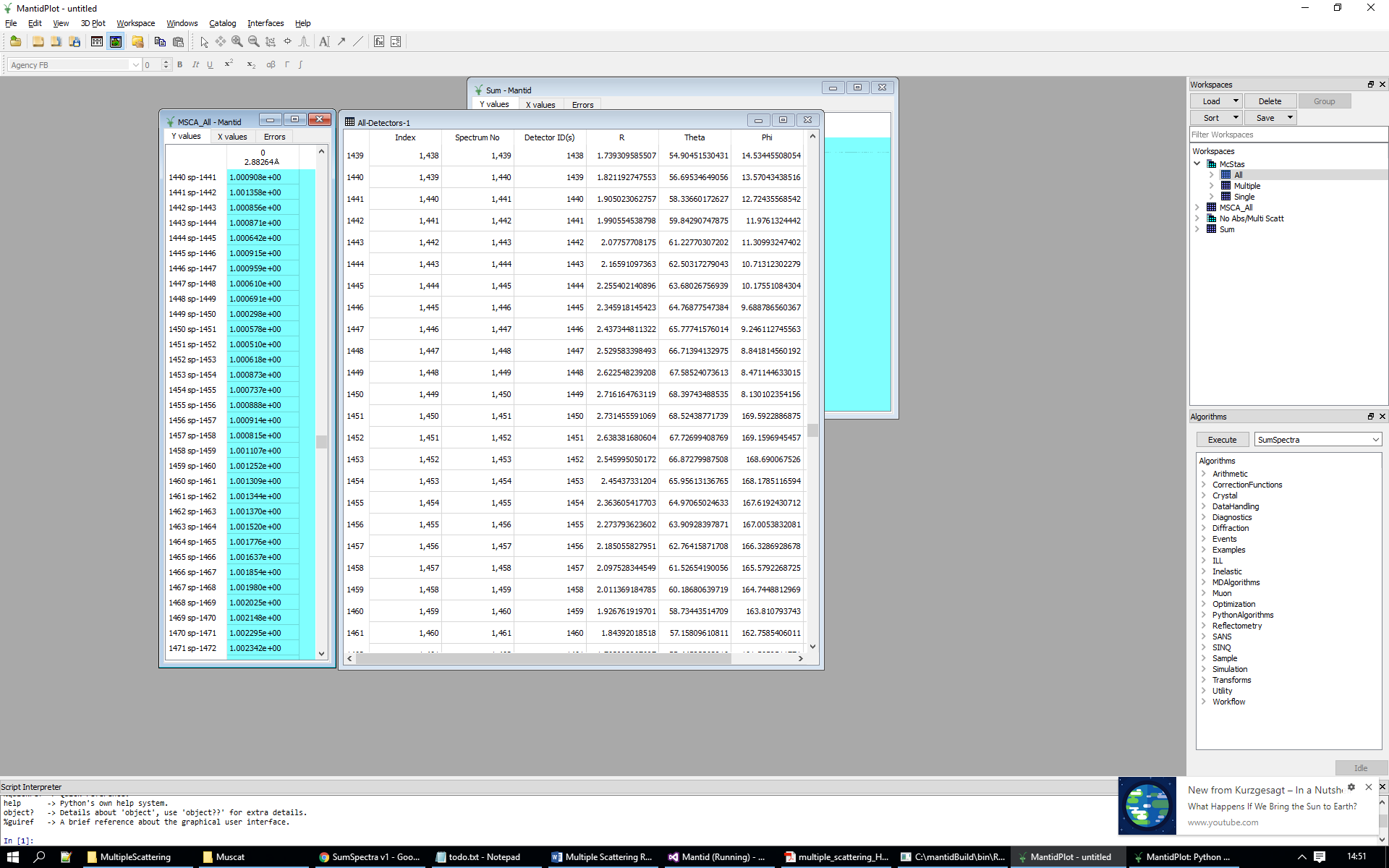


Figure 6 – (Left) a random selection of the generated correction factors for each detector, all divided by a common ratio and showing little fluctuation around 1, this is the case throughout the column. (Right) The corresponding information about the detectors showing there are differences in respective positions as we move down the list.

# Future development

## Carpenter out-of-plane cylinder absorption improvements

The Carpenter method for calculating absorption follows the approach taken in [4] which is a special case of a more general solution and therefore does not apply to cases where scattering is oblique to the scattering plane. Generalized attenuation correction factor from cylindrical targets [5] was printed as a follow up to the paper which put forward the initial algorithm and provides details of how it can be extended by the introduction of more parameters and introducing yet more expansion approximations.

For a full understanding of the method the original paper should be referred to; this section aims to simply give a rough summary of the steps that would be needed to actually implement the generalisation. Of course if it were possible to somehow incorporate the logic of the Cylinder Absorption algorithm into the Carpenter multiple scattering correction instead then this would make attempting to extend Carpenter in this way superfluous because CA is already tried and tested and we could avoid unnecessary repetition of functionality within the Mantid code. Additionally, there is no guarantee that implementing the extension outlined here will even considerably improve the accuracy of the output beyond what was presented in the earlier comparison. With these considerations in mind let us now examine the general expression for the attenuation factor:



Where (note the binomial coefficients).



The function Z is the same as before Z_{mn}(\theta) = \sum_{s=0}^\infty c_{s}(m,n)cos(s\theta) except that now it takes a different argument, chi which is defined below. We should take l/R and l’/R to be equal to 1 which gives a good agreement to the true value for Sigma R less than around 2.0 above this is still not bad.

Now in order to get somewhere in Mantid we must first calculate the incoming and scattered neutron path vector. The incoming vector need only be calculated once and then will not change throughout the solution. It is the difference between the sample and source vectors:

https://latex.codecogs.com/gif.latex?%5Coverrightarrow%7B%5COmega_i%7D%20%3D%20%5B%5Ctextup%7Bsample%7D%20%5Crightarrow%20%5Ctextup%7Bgetpos%28%29%7D%5D%20-%20%5B%5Ctextup%7Bsource%7D%5Crightarrow%20%5Ctextup%7Bgetpos%28%29%7D%5D

The detector vector then needs to be calculated in turn for each spectrum in the workspace:

C:\Users\ezz64382\Downloads\CodeCogsEqn.gif

Once we have these vectors we may then work out all the relevant angle relations needed in the attenuation expression above:

* The incident angle https://latex.codecogs.com/gif.latex?%5Cbeta_i%20%3D%20%5Carcsin%7B%5Coverrightarrow%7B%5COmega_i%7D%20%5Ccdot%20%5Cwidehat%7Bz%7D%7D
* The scattering angle https://latex.codecogs.com/gif.latex?%5Cbeta_s%20%3D%20%5Carcsin%7B%5Coverrightarrow%7B%5COmega_s%7D%20%5Ccdot%20%5Cwidehat%7Bz%7D%7D
* https://latex.codecogs.com/gif.latex?%5Ccos%7B%5Cchi%7D%20%3D%20%5Cfrac%7B%5Ccos%7B%5Ctheta%7D%20-%20%5Csin%7B%5Cbeta_i%7D%5Csin%7B%5Cbeta_s%7D%20%7D%7B%5Ccos%7B%5Cbeta_i%7D%5Ccos%7B%5Cbeta_s%7D%7D
* https://latex.codecogs.com/gif.latex?%5Ccos%7B%5Ctheta%7D%20%3D%20%5Cvec%7B%5COmega_i%7D%5Ccdot%5Cvec%7B%5COmega_s%7D

## Multiple scattering inside infinite plane slab

Another ‘low hanging fruit’ in terms of multiple scattering geometries is the infinite plane slab which could be used to model a rectangular sample. [14] and [15] are good starting points for a look into multiple scattering in a plane slab. Since the majority of samples are cylindrical and tend not to be rectangular a solution of this kind might find limited applicability but it is mentioned here anyway as a point of interest

## McStas S(q,w) procedure

Monte Carlo methods seem to be the best approach to generalising multiple scattering to any geometry because they allow us to get around the problem described in the introduction of needing to intensively compute multiple volume integrals. This is the approach followed by many of the more recent papers (see for example Copley [16] or Mancinelli [17]) and that followed by the Muscat algorithm.

This is where McStas comes in as a potentially valuable tool for implementing a Monte Carlo style procedure to correct multiple scattering. McStas has the benefit of established ease in defining the geometry of samples and the ability using the “Union” components to simulate multiple scattering for the complicated geometries of not only the sample but also its surrounding container. With some fine-tuning to improve intercommunication between Mantid it could make for a powerful simulation engine within a correction algorithm.

The McStas sample Isotropic\_Sqw is designed to handle elastic/inelastic, coherent and incoherent scattering – depending on the input S(q,w) – with multiple scattering and absorption, and is able to handle complex geometries, simulating any closed non-convex polyhedra [18]. This is, in theory, the ideal component for our uses however there have been some concerns raised about the accuracy of the sample. When comparing the ratio of multiple to single scattering in superfluid helium, one user found the ratio to be about twice that expected from semi-analytical calculations.

An accurate way of modelling a sample with a general scattering function is an integral part of the entire procedure and so it is important that more testing be carried out on this sample as well as an examination of its source code and logic to ensure that it is reliable. Vanadium would once again be a good preliminary sample material to use as a test because of its simplicity and the pre-existing algorithms available to verify the outputs. From there more complicated samples could be tested.

From the Mantid perspective, a reliable and accurate way of producing tables of S(q,w) from workspaces is needed. There exists the SofQW algorithm in Mantid but this algorithm is for use only by inelastic instruments and so could not be used on elastic vanadium experiments for example. Additionally the accuracy of the algorithm has not been examined and this would be a useful exercise if it were to be utilised inside our algorithm.

Going forward with this procedure an accurate and reliable way of producing S(q,w) tables for a variety of data as well as trust is the Isotropic\_Sqw sample used by McStas are currently the biggest obstacles to be overcome. Reading instrument data from the Mantid workspace into a format that McStas is able to read and simulate is also a poignant issue. Mantid could be enhanced to generate McStas instrument files directly from all the IDF descriptions included in the Mantid framework however this is unlikely to be unnecessary as work is currently being done to develop an improved and standard instrument geometry format based on Object File Format (OFF) which is already supported by McStas. If these developments are successful this would greatly improve one of the issues with communicating between the two pieces of software.

# Appendix

## Cylinder absorption comparison python script (Polaris 95598)

# USER SETUP - Note this script takes 5-15 minutes for multiple scattering corrections

cylinder\_radius = 0.4 # Radius is in cm

cylinder\_height = 4.0 # Height is in cm

cylinder\_position = [0., 0., 0.]

geometry\_json = {'Shape': 'Cylinder', 'Height': cylinder\_height, 'Radius': cylinder\_radius, 'Center': cylinder\_position}

material\_json = {'ChemicalFormula': 'V'}

# Essential files required : -

# Offset file:

cal\_file\_path = r"C:\...\cycle\_16\_3\_silicon\_all\_spectra.cal"

# Grouping file:

grouping\_file\_path = r"C:\...\Master\_copy\_of\_grouping\_file\_with\_essential\_masks.cal"

#----------------------------------------------------------------------------------------------------#

# Load vanadium and set the sample

Load(Filename= 'POL95598.nxs', OutputWorkspace='POL95598')

NormaliseByCurrent(InputWorkspace='POL95598', OutputWorkspace='POL95598')

SetSample(InputWorkspace='POL95598', Geometry=geometry\_json, Material=material\_json)

# Load background count

Load(Filename='POL95597.nxs', OutputWorkspace='POL95597')

NormaliseByCurrent(InputWorkspace='POL95597', OutputWorkspace='POL95597')

# subtract the background from the vanadium

Minus(LHSWorkspace='POL95598', RHSWorkspace='POL95597', OutputWorkspace='POL95598')

DeleteWorkspace('POL95597')

# get rid of 'negative' counts at very low TOF

CropWorkspace(InputWorkspace='POL95598', OutputWorkspace='POL95598', XMin=750, XMax=20000)

AlignDetectors(InputWorkspace='POL95598', OutputWorkspace='POL95598', CalibrationFile=cal\_file\_path)

# the first 55 detectors are unused detectors or monitors on Polaris

MaskDetectors(Workspace='POL95598', SpectraList='1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55')

# Create workspaces in the correct units

ConvertUnits(InputWorkspace="POL95598", OutputWorkspace="POL95598\_Carpenter", Target='Wavelength')

CloneWorkspace(InputWorkspace="POL95598\_Carpenter", OutputWorkspace="POL95598\_CA")

ConvertUnits(InputWorkspace="POL95598", OutputWorkspace="POL95598\_Mayers", Target='TOF')

#Added flag to MultipleScatteringCylinderAbsorption to disable multiple scattering corrections

MultipleScatteringCylinderAbsorption(InputWorkspace="POL95598\_Carpenter", OutputWorkspace="POL95598\_Carpenter", CylinderSampleRadius=cylinder\_height, MultipleScattering=False)

# CylinderAbsorption corrections

ca\_ws\_abs\_factors = CylinderAbsorption(InputWorkspace="POL95598\_CA", CylinderSampleHeight=mayers\_cylinder\_height, CylinderSampleRadius=cylinder\_radius)

POL95598\_CA = Divide(LHSWorkspace="POL95598\_CA", RHSWorkspace=ca\_ws\_abs\_factors)

# Mayer's correction

MayersSampleCorrection(InputWorkspace="POL95598\_Mayers", MultipleScattering=False, OutputWorkspace="POL95598\_Mayers")

# to dSpacing + focussing

for ws in ["POL95598","POL95598\_Carpenter", "POL95598\_Mayers", "POL95598\_CA"]:

loop\_ws = ConvertUnits(InputWorkspace=ws, OutputWorkspace=ws, Target='dSpacing')

DiffractionFocussing(InputWorkspace=loop\_ws, OutputWorkspace=loop\_ws, GroupingFileName=grouping\_file\_path)

division\_list = []

division\_list.append(mtd['POL95598\_Mayers']/mtd['POL95598\_CA'])

division\_list.append(mtd['POL95598\_Carpenter']/mtd['POL95598\_CA'])

plot(division\_list, 3)

## Absorption and multiple scattering python script (Polaris 95598)

# [USER SETUP / Load vanadium and background count as in Appendix A]

# Multiple Scattering Cylinder Absorption correction

ConvertUnits(InputWorkspace='POL95598', OutputWorkspace='POL95598\_Carpenter', Target='Wavelength')

MultipleScatteringCylinderAbsorption(InputWorkspace='POL95598\_Carpenter', OutputWorkspace='POL95598\_Carpenter', CylinderSampleRadius=0.4)

# Mayers correction

ConvertUnits(InputWorkspace='POL95598', OutputWorkspace='POL95598\_Mayers', Target='TOF')

MayersSampleCorrection(InputWorkspace='POL95598\_Mayers', MultipleScattering=True, OutputWorkspace='POL95598\_Mayers')

# The correction takes a long time. Create a backup copy incase irreversible changes are made so we don't need to re compute everything

CloneWorkspace(InputWorkspace='POL95598\_Mayers', OutputWorkspace='POL95598\_MayersCopy')

# Group workspaces and put to dSpacing + focussing

GroupWorkspaces(InputWorkspaces='POL95598, POL95598\_Mayers, POL95598\_Carpenter', OutputWorkspace='POL')

wsgroup = mtd['POL']

wsgroup\_names = wsgroup.getNames()

# Seems to introduce overall scale factor. Multiply by ~ 10 to get data all on same scale

Scale(InputWorkspace='POL95598\_Mayers',OutputWorkspace='POL95598\_Mayers',Factor=10)

for ws in wsgroup\_names:

loop\_ws = ConvertUnits(InputWorkspace=ws, OutputWorkspace=ws, Target='dSpacing')

DiffractionFocussing(InputWorkspace=loop\_ws, OutputWorkspace=loop\_ws, GroupingFileName=grouping\_file\_path)

Mayers\_Carpenter = Divide(LHSWorkspace='POL95598\_Carpenter', RHSWorkspace='POL95598\_Mayers')

plot(Mayers\_Carpenter, 3)

## Incoherent vanadium sample McStas instrument file

DEFINE INSTRUMENT Vanadium3()

DECLARE

%{

int multi\_flag ;

int single\_flag ;

%}

TRACE

COMPONENT Origin = Progress\_bar(percent=5)

AT (0,0,0) ABSOLUTE

EXTEND %{

single\_flag = multi\_flag = 0;

%}

COMPONENT sourceMantid = Source\_simple(radius=0.001, dist=1,

focus\_xw=0.001, focus\_yh=0.001, E0=5, dE=0.01)

AT (0, 0, 0) RELATIVE Origin

COMPONENT sampleMantid = Incoherent(radius=0.002,

yheight=0.015, focus\_r=0, pack=1, target\_x=0,

target\_y=0, target\_z=1, f\_QE=0, gamma=0)

AT (0, 0, 1) RELATIVE sourceMantid

EXTEND

%{

if (SCATTERED == 1) single\_flag =1 ;

if (SCATTERED > 1) multi\_flag =1 ;

%}

COMPONENT nD\_Mantid\_0 = Monitor\_nD(

options ="mantid square, x limits=[-2.5 2.5] bins=128 y limits=[-2.5 2.5] bins=128, neutron pixel t limits [0.002,0.005] list all neutrons, file=all",

xwidth = 5, yheight = 5, restore\_neutron = 1)

AT (0, 0, 1) RELATIVE sampleMantid

COMPONENT nD\_Mantid\_1 = Monitor\_nD(

options ="mantid square, x limits=[-2.5 2.5] bins=128 y limits=[-2.5 2.5] bins=128, neutron pixel t limits [0.002,0.005] list all neutrons, file=single",

xwidth = 5, yheight = 5, restore\_neutron = 1)

WHEN (single\_flag ==1)

AT (0, 0, 1) RELATIVE sampleMantid

END

## McStas compilation/simulation batch script for Windows

NOTE: For this to work the C flags must be correctly set to the installed Nexus library paths in the McStas configuration file. This can be done through the McStas GUI, for full details see [12]. First the McStas environment is setup and then, provided there is a folder containing an instrument file of the same name which is given by the input, the McStas and Mantid workflow is carried out and a 10e6 simulations are performed and placed in a folder called output, overwriting any previous data that were stored there:

@cd %USERPROFILE%

@CALL C:\\mcstas-2.4.1\\bin\\mccodeenv.bat

@ECHO OFF

@ECHO McStas environment shell:

@ECHO OFF

@ECHO OFF

@mcstas -v

@ECHO OFF

set "directory=C:\...\McStas"

set "file\_name=Vanadium"

set /p "file\_name=Enter name of .instr file. press [ENTER] for default [%file\_name%]: "

cd %directory%\%file\_name%

if exist output rmdir /q /s output

cmd /c "mcrun -c -n0 %file\_name%.instr”

cmd /c “mcdisplay-pl %file\_name%.instr --format=Mantid -n0 –complete

cmd /c “mcrun %file\_name%.instr -n1e6 --format=Nexus -d output"

## Mantid McStas loader python script

# USER SETUP

# Radius is in cm

cylinder\_radius = 0.1

# Details of cylinder geometry

cylinder\_height = 1.5

cylinder\_position = [0., 0., 0.]

geometry\_json = {'Shape': 'Cylinder', 'Height': cylinder\_height, 'Radius': cylinder\_radius, 'Center': cylinder\_position}

material\_json = {'ChemicalFormula': 'V'}

# Set folder path:

folder\_name = "VanadiumNoDet"

folder\_path = r"C:\Users\ezz64382\Desktop\McStas\\" + folder\_name + "\\"

#----------------------------------------------------------------------------------------------------------------------#

# Load McStas data

LoadMcStas(Filename=folder\_path + r"\output\mccode.h5", OutputWorkspace='McStas')

# Scale data from seconds to microseconds

ScaleX(InputWorkspace='McStas',Factor=1e6, OutputWorkspace='McStas')

# Manually set instrument to McStas generated IDF file

LoadInstrument(Workspace='McStas', Filename=folder\_path + folder\_name + '.instr.xml', RewriteSpectraMap=True)

# Rename workspaces for ease

RenameWorkspace(InputWorkspace='all.t\_I\_McStas', OutputWorkspace='All')

RenameWorkspace(InputWorkspace='multi.t\_I\_McStas', OutputWorkspace='Multiple')

RenameWorkspace(InputWorkspace='single.t\_I\_McStas', OutputWorkspace='Single')

# Manually put units in time of flight

wsgroup = mtd['McStas']

wsgroup\_names = wsgroup.getNames()

for wsname in wsgroup\_names:

ws = mtd[wsname]

# Set sample for each workspace

SetSample(InputWorkspace=ws, Geometry=geometry\_json, Material=material\_json)

ws.getAxis(0).setUnit('TOF')

# Cylinder absorption correction on Single without multiple scattering to get our idealised workspace

ConvertUnits(InputWorkspace='Single', Target='Wavelength', OutputWorkspace='Ideal')

CylinderAbsorption(InputWorkspace='Ideal', CylinderSampleHeight=cylinder\_height, CylinderSampleRadius=cylinder\_radius, OutputWorkspace='attfac')

Divide(LHSWorkspace='Ideal', RHSWorkspace='attfac', OutputWorkspace='Ideal')

DeleteWorkspace('attfac')

# Carpenter

ConvertUnits(InputWorkspace='All', Target='Wavelength', OutputWorkspace='Carpenter')

MultipleScatteringCylinderAbsorption(InputWorkspace='Carpenter', OutputWorkspace='Carpenter', CylinderSampleRadius=cylinder\_radius)

# Apply Mayers correction and put in units of Wavelength for comparison

MayersSampleCorrection(InputWorkspace='All', MultipleScattering=True, OutputWorkspace='Mayers')

ConvertUnits(InputWorkspace='Mayers', Target='Wavelength', OutputWorkspace='Mayers')

# There seems to be a scaling factor of 3 coming from somewhere

Scale(InputWorkspace='Mayers', Factor=0.333, OutputWorkspace='Mayers')

# Plot the two workspaces alongside the ideal one

GroupWorkspaces(InputWorkspaces='Ideal,Carpenter,Mayers', OutputWorkspace='No Abs/Multi Scatt')

ConvertUnits(InputWorkspace='All', Target='Wavelength', OutputWorkspace='AllWL')

plot('No Abs/Multi Scatt',0)

# Divide the correct results by the actual result to see if we get a horizontal straight line plot

Carpenter\_Ideal = Divide('Carpenter','Ideal')

Mayers\_Ideal = Divide('Mayers','Ideal')

Carpenter\_All = Divide('Carpenter','AllWL')

Mayers\_All = Divide('Mayers','AllWL')

GroupWorkspaces(InputWorkspaces='Carpenter\_Ideal,Mayers\_Ideal,Carpenter\_All,Mayers\_All', OutputWorkspace='Ratios')

plot('Ratios',0)

# References

1. Lindley, E.J., & Mayers, J. Cywinski, R. (Ed.). (1988). Experimental method and corrections to data. United Kingdom: Adam Hilger. - <https://inis.iaea.org/search/search.aspx?orig_q=RN:20000574>
2. V.F. Sears (1975): Slow-neutron multiple scattering, Advances in Physics, 24:1, 1-45
3. A.K.Soper, W.S.Howells and A.C.Hannon ATLAS - Analysis of Time-of-Flight Diffraction Data from Liquid and Amorphous Samples Rutherford Appleton Laboratory Report (1989): RAL-89-046
4. J.M. Carpenter Attenuation Correction Factor for Scattering from Cylindrical Targets Review of Scientific Instruments 40.4 (1969): 555. doi: 10.1063/1.1684003
5. D.F.R. Mildner, J.M. Carpenter, and C.A. Pelizzari Generalized Attenuation Correction Factor for Scattering from Cylindrical Targets Review of Scientific Instruments 45.4 (1974): 572. doi: 10.1063/1.1686687
6. D.F.R. Mildner and J.M.Carpenter Improvements to the Chebyshev Expansion of Attenuation Correction Factors for Cylindrical Samples. J Appl Crystallogr 23.5 (1990): 378–386 doi: 10.1107/S0021889890005258
7. Rouse, K. D., Cooper, M. J., York, E. J. & Chakera, A. (1970). Acta Cryst. A26, 682-691.
8. http://www.mcstas.org; Willendrup P. Farhi E. Lefmann K. McStas 1.7 - A New Version of the Flexible Monte Carlo Neutron Scattering Package, Physica B: Condensed Matter. Volume 350, Issues 1–3, Supplement, 15 July 2004, Pages E735–E737 / doi:10.1016/j.physb.2004.03.193
9. Willendrup, P.; Farhi E.; Knudsen E.; Filges U.; Lefmann K; McStas: past, present and future.
10. Journal of Neutron Research 17, 2014 pp. 35-43 / http://content.iospress.com/articles/journal-ofneutron-research/jnr004
11. Nielsen., T.R. et al., McStas and Mantid integration, Journal of Neutron Research, vol. 18, no. 2-3, pp. 61-77, 2015 DOI: 10.3233/JNR-160026 [arXiv]
12. McStas and Mantid; <https://github.com/McStasMcXtrace/McCode/wiki/McStas-and-Mantid>
13. M. W. Johnson, AERE Report R7682 (1974)
14. G.M. Vineyard, Phys. Rev. 96 (1954) 93
15. A.K. Soper, Nucl. Instr. and Meth. 212 (1983) 337
16. Copley J R D, Verkerk P, Rowe J M Comp. Phys. Commun. 1986 40 337
17. Mancinelli R (2012) Multiple neutron scattering corrections. Some general equations to do fast evaluations. J Phys, Conf Ser 340:012033
18. The Isotropic\_Sqw component; <http://www.mcstas.org/download/components/samples/Isotropic_Sqw.html>

1. Experimenting with some workspace plots and C++ code, isolating solely the multiple scattering corrections (as opposed to the absorption corrections as done in the previous section) verified this explanation although unfortunately no images or code for this part of the investigation were retained. For completeness and it would have been beneficial, time permitting, to reproduce and include this evidence. [↑](#footnote-ref-1)
2. Absorption is still simulated in the McStas sample and this must be corrected from this workspace as well to allow direct comparison [↑](#footnote-ref-2)