

Università degli studi di Roma
“TOR VERGATA”



**Simulation of the VESUVIO eV neutron
spectrometer using Monte Carlo transport
codes**

Laurea Magistrale in Fisica

Candidato:
Alex Di Giulio

A.A. 2018/2019

Relatore Interno:

Prof. Roberto Senesi

Relatore Esterno:

Dott. Giovanni Romanelli

Learn from me, if not by my precepts, at least by my example, how dangerous is the acquirement of knowledge and how much happier that man is who believes his native town to be the world, than he who aspires to become greater than his nature will allow.

Dr Victor Frankenstein

Frankenstein; or, the Modern Prometheus

Mary Shelley

Contents

I	Introduction	1
1	Neutron Transport Codes: application	2
2	Neutron and Matter	5
2.1	Introduction	5
2.1.1	A simple interaction	7
2.2	Neutron Scattering	9
2.2.1	Coherent and incoherent scattering	13
2.2.2	Correlation functions in nuclear scattering	14
2.3	Neutron Compton Scattering	15
2.4	Y Scaling and Neutron Compton Profile	19
3	The VESUVIO spectrometer	20
3.1	Foil Cycling Technique	23
3.2	Double Difference Method	24
3.3	Data Analysis	25
3.3.1	Correction to the Impulse Approximation	30
4	Brief introduction to McStas	31
4.1	The instrument file	33
II	The VESUVIO spectrometer in McStas	35
5	Collimation	36
5.1	McStas Model	37
5.2	Collimation results	38

6	Sample	42
6.1	PowderN	43
6.2	Isotropic_SQW	46
7	The Impulse Approximation Component	49
7.1	Special Functions	62
7.2	BackScattering Results for D2O	66
8	Conclusion	71
A	SQW Algoritm	73
B	Gold Foil	75
B.1	Absorption foils	75
B.1.1	Three filter_gen	76
B.1.2	Single Filter_Gen	79
B.1.3	Generating the final energy inside the sample component	79
B.1.4	Performance Comparison	79
C	Frontscattering and γ emulation	82
C.1	McStas Forward Model	83
C.2	Preliminary Results	85
D	Other results	88
D.1	Lead	88
D.2	Vanadium	89
E	Variables and abbreviations	91
	Reference	92

Part I

Introduction

Chapter 1

Neutron Transport Codes: application

In any branch of science, in modern era, it is often not trivial to see, from the results of an experiment, the analytical model predicted or to get the physical insight hiding behind them. Scientists need to develop new techniques and refine year by year experimental set-ups in order to get rid off background and unphysical corrections to the real data. More problems arise where is physically impossible to remove the backgrounds or, in more simple cases, predict complex interactions.

In this work we are going to validate the reliability of a new tool of analysis for neutron scattering experiments. There are already a numerous set of software available at the moment but in most case they are not optimized exclusively on neutron transport. McStas [1, 2, 3] is a program package relatively new (1999) based on a meta-language designed ray-tracing calculations for neutrons. The program provides a consistent library of beam line components that can be easily arranged to reproduce accurately modern beam lines. The main limitation is that this software is based on thermal neutron scattering, it was originally developed to simulate a triple-axis spectrometer, so we are extending the energy range of the simulations trying to replicate a deep inelastic neutron scattering experiment.

The most valuable property of the neutrons is their neutral charge, that permits to such particles to ignore the Coulomb interaction and penetrate deep inside materials. This is, of course, a limitation as well: no Coulomb interaction implies difficult manipulation or confinement/shielding of the particles. Neutrons are usually emitted isotropically and need to be deflected by subsequent scattering on proper materials in order to guide them along beam lines. After hitting a sample they scatter again and can interact with many parts

of an experimental environment before reaching the detector. This adds many different background contributions and that cannot be predicted and corrected analytically, at least not all of them.

The first step towards a good experiment design is to choose wisely environment materials. For example: the sample container contribution to the signal needs to be relatively low; depending on the experiment is often used Vanadium or Aluminum. For neutrons diffraction experiments the *coherent* cross section is taken into account (see ch. 2.2.1) and the Vanadium has a *coherent* cross section of 0.0184 b, it is transparent to neutrons with energy of the order of 25 meV. For deep inelastic neutron scattering experiments the Aluminum is preferable because one use instead the *total* cross section that is 1.503 b.

The shielding material, on the other hand, need to have strongly interacting properties. In most cases we use a mixture of different nuclei. Light-weight nuclei (like hydrogen) can thermalize neutrons and lower their energy; then, high-absorbing nuclei capture such slowed neutrons (like boron, cadmium and gadolinium).

Even considering the best materials choice in every aspect and components, there are still numerous interaction with basically everything around the sample. At this point we need some help to handle complex geometries, we can not describe analytically every single component in the blockhouse, but we can delegate this boring and complex calculations to computers.

Devoted to describing neutron motion in media and the corresponding laws, *neutron transport theory* is the scientific framework at the base of transmutation and activation reactions, among other topics.[4]

There are two methods for neutron transport calculations: the Monte Carlo method and the deterministic method. The Monte Carlo method is a numerical method based on probability and statistical theories. It is based on the concept of random sampling, we do not need to calculate every possible path taken by the neutron to reach the detector but only a subset of this set of path. This subset of random paths, however, needs to be uniform distributed. It can describe the characteristics of randomly moving particles and the process of physical experiments. In contrast, in the deterministic method, a group of mathematical-physical equations is first built up to explain the physical characteristics of the target system. Then, by discretizing the variables including direction, energy, space and time in these equations, an approximate solution can be obtained with numerical calculation.

Referring to figure 1.1 one can understand the different approach. Suppose one wants to calculate the ratio between the circle area (gray) and the square area (white). In the deterministic case (right) one could subdivide N times the total area and counts how many

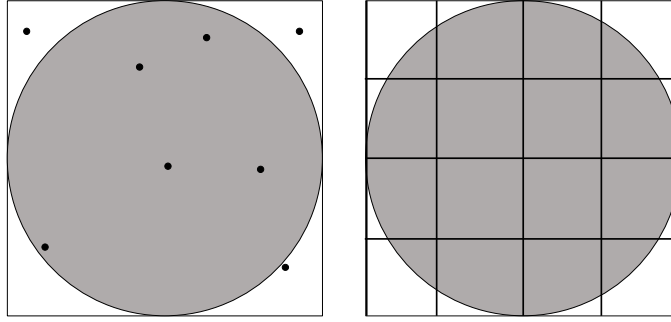


Figure 1.1: Schematic approach difference between a Monte Carlo measure (left) and a deterministic measure (right).

squares are completely gray. Dividing this number by the total number of squares we obtain an estimation of π . Solving the same task using a Monte Carlo algorithm is done by generating N random points inside the main square from a uniform distribution. Calculating the ratio

$$\lim_{N \rightarrow \infty} \frac{\text{Points inside the circle}}{N} = 4\pi \quad (1.1)$$

for $N \rightarrow \infty$ we obtain an estimation of π . The two approach reach the same result but, based on the complexity of the geometries, the time needed to obtain a good estimation can be different. To our case the Monte Carlo technique is the reasonable solution.

In this Thesis we are going to lay the foundation for a new simulation tool, developed inside the McStas package, that can help instrument scientists and researchers to analyze and optimize epithermal neutron scattering experiments performed on the VESUVIO electron volt spectrometer. Until now there was only one code that could simulate the multiple-scattering contribution on VESUVIO, the DINSMS code by Mayers et al [5]. The difficulty to simulate such instrument lies in the various branches of physics involved into a measure. Both nuclear and condensed-matter physics need to be handled inside the code and most of the software package available (like MCNPX[6]) are heavily optimized for nuclear interactions. McStas is becoming a standard tool for thermal neutron spectroscopy and most instruments¹ of this type can be easily simulated [7, 8, 9]. In this work we tried to cover the lack of **components** (see ch. 4) that can simulate electron Volt neutrons, writing a new component that uses impulse approximation to describe neutrons interaction. This is a step forward to correct multiple-scattering contribution in the measure, since the DINSMS code can only simulate a homogeneous mixture built from the materials of sample and its container. McStas, on the other hand, can handle distinct geometries and, in future, the multiple scattering between the geometries.

¹A consistent set of already implemented instruments can be found in <http://www.mcstas.org/download/components/>

Chapter 2

Neutron and Matter

2.1 Introduction

Before presenting the computational approach we recall in this chapter some useful and basic concepts about the neutron-matter interaction.

Basic properties of neutron	
mass	$1.675 \times 10^{-27} \text{ kg}$
charge	0
spin	$\frac{\hbar}{2}$
magnetic dipole moment	$-1.913\mu_N$

Most of our discussion is based on the cross-section concept. It characterizes the probability that a particular interaction will occur. We can define this quantity depending on the interaction under study: coherent cross-section, incoherent cross-section, neutron-gamma capture cross-section and so on. The cross-section is typically denoted σ and measured in units of area. The standard unit for measuring a nuclear cross section is the *barn*, which is equal to 10^{-28} m^2 or 10^{-24} cm^2 . The concept of a nuclear cross section can be understood physically in terms of *characteristic target area*, where a larger (fig 2.1) area means a larger probability of interaction.

A scattering event of a neutron by a sample is generally characterized by a change in its momentum and energy. In fig 2.2 we can see a scheme describing an incident particle with a wavevector \mathbf{k}_i and angular frequency ω_i . After the scattering event the particle has a new wavevector and angular frequency \mathbf{k}_f and ω_f . One can define the momentum transfer and the energy transfer:

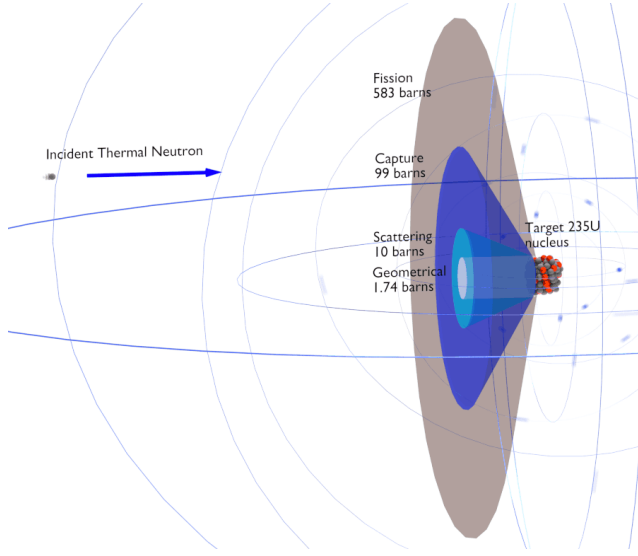


Figure 2.1: Graphical detail of various cross section of ^{235}U . The cross section are referred to thermal neutron scattering.

$$\hbar \mathbf{k}_i - \hbar \mathbf{k}_f = \hbar \mathbf{q} \quad (2.1)$$

where $\mathbf{q} = \mathbf{k}_i - \mathbf{k}_f$, and

$$E = \hbar \omega \text{ where } \omega = \omega_i - \omega_f \quad (2.2)$$

Consider a monochromatic beam of neutrons, all with the same energy E , incident on a target (fig 2.2). Suppose we set up a neutron counter and measure the number of neutrons scattered in a given direction as a function of their energy, E' . We define the *double differential cross-section* by

$$\frac{d^2\sigma}{d\Omega dE'} = \frac{\left(\begin{array}{l} \text{Number of neutrons scattered per second into a small solid angle } d\Omega \text{ in} \\ \text{the direction } \theta, \phi \text{ with final energy between } E' \text{ and } E' + dE' \end{array} \right)}{\Phi d\Omega dE'} \quad (2.3)$$

where Φ is the flux of the incident neutrons. Suppose we do not analyze the energy of the scattered neutrons, but simply count all the neutrons scattered into the solid angle $d\Omega$ around the direction θ, ϕ . The cross-section corresponding to these measurements, the *differential cross-section* is defined by

$$\frac{d\sigma}{d\Omega} = \frac{(\text{number of neutrons scattered per second into } d\Omega \text{ in the direction } \theta, \phi)}{\Phi d\Omega} \quad (2.4)$$

The total scattering cross-section is defined by the equation

$$\sigma_{tot} = \frac{(\text{total number of neutrons scattered per second})}{\Phi} \quad (2.5)$$

From the previous definitions one has:

$$\begin{aligned}\frac{d\sigma}{d\Omega} &= \int_0^\infty \frac{d^2\sigma}{d\Omega dE'} \\ \sigma_{tot} &= \int \left(\frac{d\sigma}{d\Omega} \right) d\Omega\end{aligned}\tag{2.6}$$

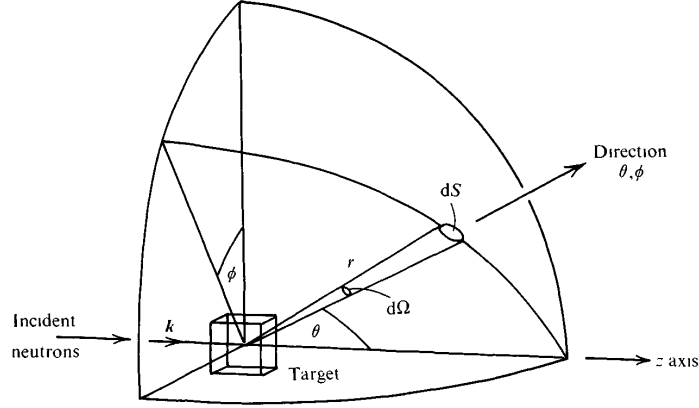


Figure 2.2: A schematic representation of a particle being scattered by a sample. Figure adapted from ref [10]

Before going deep into the physics of a single reaction (in this thesis we are going to investigate mainly Compton Scattering in ch. 2.3) we can consider some general derivations used in this work.

2.1.1 A simple interaction

Consider a thermal neutron, i.e. a neutron with an energy comparable with $k_B T \simeq 25 \text{ meV}$. The neutron wavelength is of the order of 10^{-10} m , much larger than typical distance of interaction by nuclear forces which cause the scattering (from 10^{-14} m to 10^{-15} m). In this scenario the scattering, analyzed in terms of partial waves [11], comes entirely from the S waves (azimuthal quantum number $l = 0$), which are spherically symmetric. We will consider the origin of the axes to be at the position of the rest nucleus and the z axis along the direction of \mathbf{k} , the wavevector of the incident neutrons. We can represent the incident neutrons by a plane-wave wavefunction:

$$\psi_{inc} = \exp(ikz)\tag{2.7}$$

As the scattering is spherically symmetric, the wavefunction of the scattered neutrons at the point \mathbf{r} can be written in the form

Table 2.1: Table of naming convention of neutrons

Neutrons	Energy range	Wavelength [\AA]	Velocity [m/s]
ultra cold	$\leq 300 \text{ neV}$	≥ 500	≤ 8
very cold	300 neV - 0.12 meV	52.2 – 26.1	7.5 – 152
cold	0.12 meV - 12 meV	26.1 – 2.6	152 – 1515
thermal	12 meV - 500 meV	2.6 - 0.9	1515 - 4374
epithermal	500 meV - 1keV	0.9 - 0.28	4374 - 13.8 103
intermediate	1keV - 0.8MeV		
fast	$> 0.8\text{MeV}$		

$$\psi_{sc} = -\frac{b}{r} \exp(ikr) \quad (2.8)$$

where b is a constant, independent of the angles θ and ϕ . The magnitude of the wavevector is unchanged, the energy of thermal neutrons is too small to change the internal energy of the nucleus. In this first approximation we are taking the position of the nucleus fixed, so the neutron cannot transfer kinetic energy to the nucleus. The scattering in this assumption is referred to as elastic: neutron energy is conserved. The quantity b in ψ_{sc} is known as the *scattering length*. This quantity is complex and the imaginary part is directly connected to the capture cross section. Some nuclei form a compound nucleus (original nucleus plus the neutron) with energy close to an excited state. Since the imaginary part of the scattering length corresponds to absorption, such nuclei strongly absorb neutrons. In principle, one can calculate the values of b from within nuclear models provided a detailed knowledge of all the quantum numbers of the neutron and nucleus involved. In practice, the values of b do not show any trivial dependence on the atomic and atomic mass numbers Z , A , as showed in Figure 2.3.

Following the previous definitions, we can calculate the cross-section $d\sigma/d\Omega$ for scattering from a single fixed nucleus. If v is the velocity of both incoming and scattered neutrons, the number of neutrons passing through an area $d\mathbf{S}$ per second is

$$\begin{aligned} v d\mathbf{S} |\psi_{sc}|^2 &= v d\mathbf{S} \frac{b^2}{r^2} \\ &= vb^2 d\Omega \end{aligned} \quad (2.9)$$

The flux of incident neutrons is

$$\Phi = v |\psi_{inc}|^2 \quad (2.10)$$

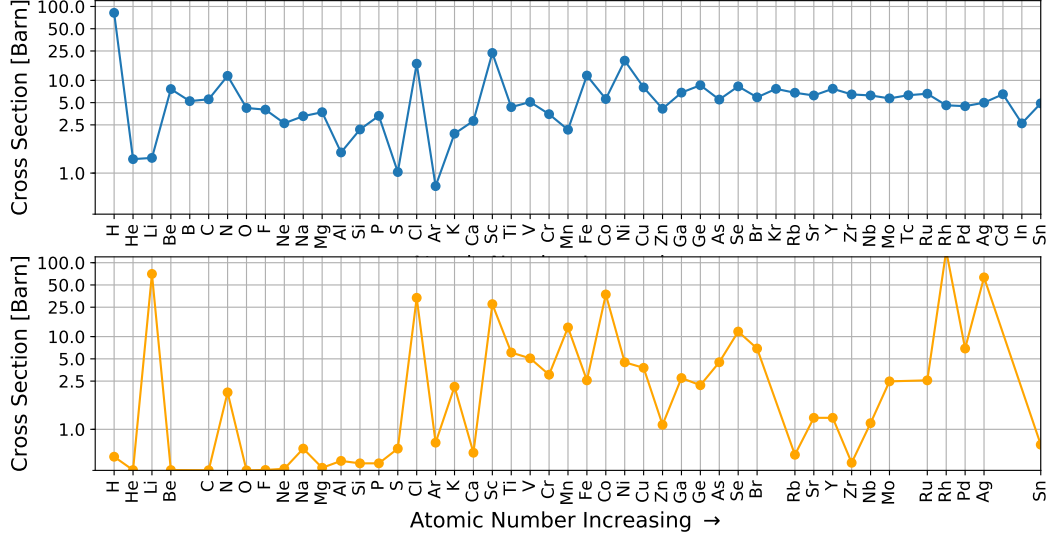


Figure 2.3: Neutron scattering (top) and absorption (bottom) cross section in function of Atomic Number up to Sn. Data is retrieved from tables on NIST Center for Neutron Research. ¹

From the definition of the cross-section

$$\frac{d\sigma}{d\Omega} = \frac{vb^2 d\Omega}{\Phi d\Omega} = b^2 \quad (2.11)$$

so we obtain

$$\sigma_{tot} = 4\pi b^2 \quad (2.12)$$

2.2 Neutron Scattering

So far, we have neglected the spin state and magnetic interaction of the neutrons. This means that a neutron state is completely described by its momentum, i.e. wavevector. We are going to derive a general expression for the double differential cross-section $d^2\sigma/d\Omega dE'$ for a specific transition of the scattering system from one of its quantum states to another. Suppose we have a neutron with wavevector \mathbf{k} incident on a scattering system in a status λ . Denote by $\psi_{\mathbf{k}}$ the neutron wavefunction and χ_{λ} the scattering system wavefunction. Suppose the neutron interacts via a potential v , the final neutron wavevector is \mathbf{k}' and the final scattering system state is λ' . If N is the number of nuclei in the scattering system we write $\mathbf{R}_j (j = 1, \dots, N)$ the position vector of the j th nucleus, and with \mathbf{r} the position of the neutron.

¹<https://www.ncnr.nist.gov/resources/n-lengths/list.html>

We start considering the single differential cross section $d\sigma/d\Omega$ representing the sum of all processes in which the state of the scattering system changes from λ to λ' and the neutron changes from \mathbf{k} to \mathbf{k}' . Namely:

$$\left(\frac{d\sigma}{d\Omega}\right)_{\lambda \rightarrow \lambda'} = \frac{1}{\Phi} \frac{1}{d\Omega} \sum_{\mathbf{k}' \text{ in } d\Omega} W_{\mathbf{k}, \lambda \rightarrow \mathbf{k}', \lambda'} \quad (2.13)$$

Where $W_{i \rightarrow f}$ is the number of transitions per second from initial to final state. This quantity can be evaluated through the *Fermi's golden rule*

$$W_{\mathbf{k}, \lambda \rightarrow \mathbf{k}', \lambda'} = \frac{2\pi}{\hbar} \rho_{k'} |\langle \mathbf{k}' \lambda' | v | \mathbf{k} \lambda \rangle|^2 \quad (2.14)$$

Where $\rho_{k'}$ is the number of momentum states in $d\Omega$ per unit energy range for neutron in the state \mathbf{k}'

$$\rho_{k'} = \frac{V}{2\pi} k' \frac{m}{\hbar^2} d\Omega \quad (2.15)$$

The matrix element is given explicitly by

$$\langle \mathbf{k}' \lambda' | v | \mathbf{k} \lambda \rangle = \int \psi_{\mathbf{k}'}^\dagger \chi_{\lambda'}^\dagger v(\mathbf{R}, \mathbf{r}) \psi_{\lambda} \chi_{\mathbf{k}} d\mathbf{R} d\mathbf{r} \quad (2.16)$$

where $d\mathbf{R} = d\mathbf{R}_1 d\mathbf{R}_2 \dots d\mathbf{R}_N$. Using the box normalization we can use as neutron function

$$\psi_{\mathbf{k}} = \frac{1}{\sqrt{V}} \exp(i\mathbf{k} \cdot \mathbf{r}) \quad (2.17)$$

using the plane wave for neutron states we can write the matrix element

$$\begin{aligned} \langle \mathbf{k}' \lambda' | v | \mathbf{k} \lambda \rangle &= \int \psi_{\mathbf{k}'}^\dagger \chi_{\lambda'}^\dagger v(\mathbf{R}, \mathbf{r}) \psi_{\lambda} \chi_{\mathbf{k}} d\mathbf{R} d\mathbf{r} \\ &= \frac{1}{V} \int e^{-i\mathbf{k}' \cdot \mathbf{r}} \chi_{\lambda'}^\dagger v(\mathbf{R}, \mathbf{r}) e^{i\mathbf{k} \cdot \mathbf{r}} \chi_{\lambda} d\mathbf{R} d\mathbf{r} \end{aligned} \quad (2.18)$$

Now the flux of the incident neutrons is the product of their density and velocity, i.e.

$$\Phi = \frac{1}{V} \frac{\hbar}{m} k \quad (2.19)$$

so we finally can write from 2.13

$$\left(\frac{d\sigma}{d\Omega}\right)_{\lambda \rightarrow \lambda'} = \frac{k'}{k} \left(\frac{m}{2\pi\hbar^2}\right)^2 |\langle \mathbf{k}' \lambda' | v | \mathbf{k} \lambda \rangle|^2 \quad (2.20)$$

considering the conservation of energy we write

$$\left(\frac{d^2\sigma}{d\Omega dE'}\right)_{\lambda \rightarrow \lambda'} = \frac{k'}{k} \left(\frac{m}{2\pi\hbar^2}\right)^2 |\langle \mathbf{k}' \lambda' | v | \mathbf{k} \lambda \rangle|^2 \delta(E_{\lambda} - E_{\lambda'} + E - E') \quad (2.21)$$

Moving to the Fourier transform we can simplify the mathematics and obtain some useful results. Consider the Fourier transform of the potential function

$$\begin{aligned}\langle \mathbf{k}'\lambda|v|\mathbf{k}\lambda\rangle &= \sum_j \int e^{-i\mathbf{k}'\cdot\mathbf{r}} \chi_{\lambda'}^\dagger v_j(\mathbf{r} - \mathbf{R}_j) e^{i\mathbf{k}\cdot\mathbf{r}} \chi_\lambda d\mathbf{R} d\mathbf{r} \\ &= \sum_j v_j(\mathbf{q}) \langle \lambda'|e^{i\mathbf{q}\cdot\mathbf{R}_j}|\lambda\rangle\end{aligned}\quad (2.22)$$

where $\mathbf{x}_j = \mathbf{r} - \mathbf{R}_j$

$$v_j(\mathbf{q}) = \int v_j(\mathbf{x}_j) e^{i\mathbf{q}\cdot\mathbf{x}_j} d\mathbf{x}_j \quad (2.23)$$

$$\langle \lambda'|e^{i\mathbf{q}\cdot\mathbf{R}_j}|\lambda\rangle = \int \chi_{\lambda'}^\dagger e^{i\mathbf{q}\cdot\mathbf{R}_j} \chi_\lambda d\mathbf{R} \quad (2.24)$$

and

$$\mathbf{q} = \mathbf{k} - \mathbf{k}' \quad (2.25)$$

Considering a single fixed nucleus in the origin in 2.22, as to have $\mathbf{R}_1 = 0$, and $\lambda' = \lambda$,

$$\begin{aligned}\langle \mathbf{k}'\lambda|v|\mathbf{k}\lambda\rangle &= \int \chi_\lambda^\dagger \chi_\lambda d\mathbf{R}_1 \int v(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r} \\ &= \int v(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r}\end{aligned}\quad (2.26)$$

inserting result in 2.20, together with $k' = k$ gives

$$\frac{d\sigma}{d\Omega} = \left(\frac{m}{2\pi\hbar^2} \right)^2 \left| \int v(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r} \right|^2 \quad (2.27)$$

Nuclear scattering of the free neutron by the nucleus is mediated by the strong nuclear force. The wavelength of thermal (several Å) and cold neutrons (up to tens of Angstroms) typically used for such investigations is 4-5 orders of magnitude larger than the dimension of the nucleus (femtometres). With these assumptions one can describe the spatial density function of the nucleus as a delta function. The potential then can be described as

$$v(\mathbf{r}) = a\delta(\mathbf{r})$$

where a is a real constant. Then

$$\int v(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r} = a \int \delta(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r} = a \quad (2.28)$$

but from 2.11 and 2.27 we obtain:

$$a = \frac{2\pi\hbar^2}{m} b \quad (2.29)$$

so at the end we can define the *Fermi pseudopotential*

$$v(\mathbf{r}) = \frac{2\pi\hbar^2}{m} b\delta(\mathbf{r}) \quad (2.30)$$

The scattering length defined in 2.8 relates to a fixed nucleus and is know as the *bound* scattering length. If the nucleus is free, the scattering must be treated in the centre-of-mass system. We can use the same result as if the nuclei were fixed, but using μ instead of m

$$\mu = \frac{mM}{m+M}$$

were M is the mass of the nucleus. The scattering length for this process is called the *free* scattering length b_f . From the pseudopotential

$$\frac{b_f}{\mu} = \frac{b}{m} \Rightarrow b_f = \frac{b}{1+m/M}$$

so, for example, for hydrogen $b_f = b/2$. Now, using the Fourier transform of the pseudopotential and inserting it into 2.21

$$\left(\frac{d^2\sigma}{d\Omega dE'} \right)_{\lambda \rightarrow \lambda'} = \frac{k'}{k} \left| \sum_j b_j \langle \lambda' | e^{i\mathbf{q} \cdot \mathbf{R}_j} | \lambda \rangle \right|^2 \delta(E_\lambda - E_{\lambda'} + E - E') \quad (2.31)$$

so using the integral representation of the δ -function we can write

$$\begin{aligned} \left(\frac{d^2\sigma}{d\Omega dE'} \right)_{\lambda \rightarrow \lambda'} &= \frac{k'}{k} \frac{1}{2\pi\hbar} \sum_{jj'} b_{j'} b_j \int_{-\infty}^{\infty} \langle \lambda | e^{-i\mathbf{q} \cdot \mathbf{R}_{j'}} | \lambda' \rangle \\ &\times \langle \lambda' | e^{iHt/\hbar} e^{i\mathbf{q} \cdot \mathbf{R}_j} e^{-iHt/\hbar} | \lambda \rangle e^{-i\omega t} dt \end{aligned} \quad (2.32)$$

In real-life experiments we do not measure the cross-section for a specific state transition $\lambda \rightarrow \lambda'$ but we are measuring the 2.3. We need to sum over all final states and than average over all the initial states. If p_λ is the probability that the scattering system is in the state λ .

$$\begin{aligned} \frac{d^2\sigma}{d\Omega dE'} &= \sum_{\lambda\lambda'} p_\lambda \left(\frac{d^2\sigma}{d\Omega dE'} \right)_{\lambda \rightarrow \lambda'} \\ &= \frac{k'}{k} \frac{1}{2\pi\hbar} \sum_{jj'} b_{j'} b_j \int_{-\infty}^{\infty} e^{-i\omega t} dt \sum_{\lambda} p_\lambda \\ &\times \langle \lambda | e^{-i\mathbf{q} \cdot \mathbf{R}_{j'}} e^{iHt/\hbar} e^{i\mathbf{q} \cdot \mathbf{R}_j} e^{-iHt/\hbar} | \lambda \rangle \end{aligned} \quad (2.33)$$

using the Heisenberg operator $\mathbf{R}_j(t)$

$$\frac{d^2\sigma}{d\Omega dE'} = \frac{k'}{k} \frac{1}{2\pi\hbar} \sum_{jj'} b_{j'} b_j \int_{-\infty}^{\infty} \langle e^{-i\mathbf{q} \cdot \mathbf{R}_{j'}(0)} e^{i\mathbf{q} \cdot \mathbf{R}_j(t)} \rangle \exp^{-i\omega t} dt \quad (2.34)$$

were we used $\langle A \rangle = \sum_{\lambda} p_\lambda \langle \lambda | A | \lambda \rangle$

2.2.1 Coherent and incoherent scattering

We can divide the contributions to the double-differential cross section in *coherent* scattering, containing interference terms from the correlations in the positions of the atoms, or *incoherent* scattering, where we can consider the single atoms as isolated entities. Consider a scattering system with a single element where the scattering length b varies from a nucleus to another owing to nuclear spin or the presence of isotopes or both. Let the value b_i occur with relative frequency f_i , i.e.

$$\sum_i f_i = 1$$

Then the average value of b for the system is

$$\langle b \rangle = \sum_i f_i b_i$$

and the average value of b^2 is

$$\langle b^2 \rangle = \sum_i f_i b_i^2$$

now consider that the system contains a large number of nuclei. Since b_j varies for each nucleus and we cannot correlate it with position \mathbf{R}_j , we need to take the average value of $b_{j'} b_j$

$$\frac{d^2\sigma}{d\Omega dE'} = \frac{k'}{k} \frac{1}{2\pi\hbar} \sum_{jj'} \langle b_{j'} b_j \rangle \int \langle j', j \rangle e^{-i\omega t} dt \quad (2.35)$$

where $\langle j', j \rangle = \langle e^{-i\mathbf{q} \cdot \mathbf{R}_{j'}(0)} e^{i\mathbf{q} \cdot \mathbf{R}_j(t)} \rangle$. On the assumption of no correlation between the b values of different nuclei

$$\langle b_{j'} b_j \rangle = \begin{cases} \langle b \rangle^2, & \text{for } j' \neq j \\ \langle b^2 \rangle, & \text{for } j' = j \end{cases} \quad (2.36)$$

We can split the summation separating the values arising from the correlation of a nucleus with itself and the one with other distinct nuclei

$$\frac{d^2\sigma}{d\Omega dE'} = \frac{k'}{k} \frac{1}{2\pi\hbar} \langle b \rangle^2 \sum_{j' \neq j} \int \langle j', j \rangle e^{-i\omega t} dt + \frac{k'}{k} \frac{1}{2\pi\hbar} \langle b^2 \rangle \sum_j \int \langle j, j \rangle e^{-i\omega t} dt \quad (2.37)$$

which can be written as

$$\begin{aligned} \frac{d^2\sigma}{d\Omega dE'} &= \frac{k'}{k} \frac{1}{2\pi\hbar} \langle b \rangle^2 \sum_{j' \neq j} \int \langle j', j \rangle e^{-i\omega t} dt \\ &+ \frac{k'}{k} \frac{1}{2\pi\hbar} (\langle b^2 \rangle - \langle b \rangle^2) \sum_j \int \langle j, j \rangle e^{-i\omega t} dt \end{aligned} \quad (2.38)$$

so finally we can group the two contributions in *coherent* cross-section and *incoherent* cross-section.

$$\begin{aligned}\left(\frac{d^2\sigma}{d\Omega dE'}\right)_{coh} &= \frac{\sigma_{coh}}{4\pi} \frac{k'}{k} \frac{1}{2\pi\hbar} \sum_{j'j} \int_{-\infty}^{\infty} \langle e^{-i\mathbf{q}\cdot\mathbf{R}_{j'}(0)} e^{i\mathbf{q}\cdot\mathbf{R}_j(t)} \rangle e^{-i\omega t} dt \\ \left(\frac{d^2\sigma}{d\Omega dE'}\right)_{inc} &= \frac{\sigma_{inc}}{4\pi} \frac{k'}{k} \frac{1}{2\pi\hbar} \sum_j \int_{-\infty}^{\infty} \langle e^{-i\mathbf{q}\cdot\mathbf{R}_j(0)} e^{i\mathbf{q}\cdot\mathbf{R}_j(t)} \rangle e^{-i\omega t} dt\end{aligned}\quad (2.39)$$

where $\sigma_{coh} = 4\pi\langle b \rangle^2$ and $\sigma_{inc} = 4\pi(\langle b^2 \rangle - \langle b \rangle^2)$

2.2.2 Correlation functions in nuclear scattering

We have found a relation between the double difference cross sections for incoherent and coherent scattering and the *thermal average* of position operator $\mathbf{R}_j(t)$. We can rewrite the equations 2.39 introducing the *correlation functions*. These function are not only useful for calculating various properties of the scattering system, but they also provide insight into the physical significance of the terms that occur in the scattering cross-section. Consider the coherent double differential cross section as defined above:

$$\begin{aligned}\left(\frac{d^2\sigma}{d\Omega dE'}\right)_{coh} &= \frac{\sigma_{coh}}{4\pi} \frac{k'}{k} \frac{1}{2\pi\hbar} \int_{-\infty}^{\infty} \sum_{j'j} \langle e^{-i\mathbf{q}\cdot\mathbf{R}_{j'}(0)} e^{i\mathbf{q}\cdot\mathbf{R}_j(t)} \rangle e^{-i\omega t} dt \\ &= \frac{\sigma_{coh}}{4\pi} \frac{k'}{k} \frac{1}{2\pi\hbar} \int_{-\infty}^{\infty} NI(\mathbf{q}, t) e^{-i\omega t} dt\end{aligned}\quad (2.40)$$

where we have defined the *intermediate scattering function*

$$I(\mathbf{q}, t) = \frac{1}{N} \sum_{j'j} \langle e^{-i\mathbf{q}\cdot\mathbf{R}_{j'}(0)} e^{i\mathbf{q}\cdot\mathbf{R}_j(t)} \rangle \quad (2.41)$$

where N is the number of nuclei in the scattering system. We next define the functions $G(\mathbf{q}, t)$ and $S(\mathbf{q}, \omega)$ by

$$\begin{aligned}G(\mathbf{r}, t) &= \frac{1}{(2\pi)^3} \int I(\mathbf{q}, t) e^{-i\mathbf{q}\cdot\mathbf{r}} d\mathbf{q} \\ S(\mathbf{q}, \omega) &= \frac{1}{2\pi\hbar} \int I(\mathbf{q}, t) e^{-i\omega t} dt\end{aligned}\quad (2.42)$$

Using the inverse relations for Fourier transform

$$\begin{aligned}I(\mathbf{q}, t) &= \int G(\mathbf{r}, t) e^{i\mathbf{q}\cdot\mathbf{r}} d\mathbf{r} \\ I(\mathbf{q}, t) &= \hbar \int S(\mathbf{q}, \omega) e^{i\omega t} d\omega\end{aligned}\quad (2.43)$$

and so

$$\begin{aligned}G(\mathbf{r}, t) &= \frac{\hbar}{(2\pi)^3} \int S(\mathbf{q}, \omega) e^{-i(\mathbf{q}\cdot\mathbf{r} - \omega t)} d\mathbf{q} d\omega \\ S(\mathbf{q}, \omega) &= \frac{1}{2\pi\hbar} \int G(\mathbf{r}, t) e^{i(\mathbf{q}\cdot\mathbf{r} - \omega t)} d\mathbf{r} dt\end{aligned}\quad (2.44)$$

$G(\mathbf{r}, t)$ is the *time-dependent pair-correlation function* of the scattering system. $S(\mathbf{q}, \omega)$ is known as the *scattering function* of the system.

Now we can separate the sum over the same nuclei position from the total sum. In this way we are defining a *self correlation functions*, the self intermediate function:

$$I_s(\mathbf{q}, t) = \frac{1}{N} \sum_j \langle e^{-i\mathbf{q} \cdot \mathbf{R}_j(0)} e^{i\mathbf{q} \cdot \mathbf{R}_j(t)} \rangle \quad (2.45)$$

from which we can obtain the *self time-dependent pair-correlation function* and the *incoherent scattering function*

$$\begin{aligned} G_s(\mathbf{r}, t) &= \frac{1}{(2\pi)^3} \int I_s(\mathbf{q}, t) e^{-i\mathbf{q} \cdot \mathbf{r}} d\mathbf{q} \\ S_i(\mathbf{q}, \omega) &= \frac{1}{2\pi\hbar} \int I_s(\mathbf{q}, t) e^{-i\omega t} dt \end{aligned} \quad (2.46)$$

Now we can relate directly the pair-correlation functions to the double differential cross section.

$$\begin{aligned} \left(\frac{d^2\sigma}{d\Omega dE'} \right)_{coh} &= \frac{\sigma_{coh}}{4\pi} \frac{k'}{k} N S(\mathbf{q}, \omega) \\ \left(\frac{d^2\sigma}{d\Omega dE'} \right)_{inc} &= \frac{\sigma_{inc}}{4\pi} \frac{k'}{k} N S_i(\mathbf{q}, \omega) \end{aligned} \quad (2.47)$$

2.3 Neutron Compton Scattering

Despite the importance of the coherent neutron scattering in this thesis we are going to focus only on the incoherent contribution. The main reason of this choice is that the VESUVIO spectrometer scientific programme mainly concerns Deep Inelastic Neutron Scattering, even if the collected data can cover more aspects and interaction (like diffraction patterns). The basic difference between elastic and inelastic scattering is that in elastic scattering the energy of neutron is conserved before and after interaction:

$$E = E' \Rightarrow k = k' \quad (2.48)$$

In inelastic scattering we need to consider that a neutron can increase ($k' > k$) or lower ($k' < k$) its energy:

$$k = \frac{2\pi}{\lambda} \neq \frac{2\pi}{\lambda'} = k' \quad (2.49)$$

the vector $\mathbf{q} = \mathbf{k} - \mathbf{k}'$ assume the general formulation:

$$q^2 = k^2 + k'^2 - 2kk' \cos(2\theta) \quad (2.50)$$

At high energies, above typical excitation energies in a condensed matter system, the double differential neutron-scattering cross section for a single atomic species is related to the

dynamic structure factor $S(\mathbf{q}, \omega)$ (we are not separating for now the coherent or incoherent contribution) via 2.47 as

$$\frac{d^2\sigma}{d\Omega dE'} = \sqrt{\frac{E'}{E}} \frac{\sigma}{4\pi} S(\mathbf{q}, \omega) \quad (2.51)$$

where we have used

$$E = \frac{\hbar^2 k^2}{2m} \Rightarrow k = \frac{\sqrt{2mE}}{\hbar}$$

In eq. 2.51: E and E' energies of the incident and scattered neutron respectively, $\hbar\mathbf{q}$ and $\hbar\omega$ the momentum and energy transfers and σ the total scattering cross section:

$$\sigma = 4\pi [|b^2| + (|b^2| - |b|^2)] = 4\pi |b|^2 \quad (2.52)$$

where b is the neutron-scattering length for the target nuclide. Note again that in this work and theory we are not considering magnetic contributions. The previous equation separate two contributions to the scattering process. As before explained (section 2.2.1) the first term describe a *coherent* process, where neutron were used in much the same way as X-ray crystallography, the latter the *incoherent*, where neutron lose information about structure and from we can extrapolate information about diffusion process or vibrational dynamics of atoms in condensed matter. In our case, with high energy transfer above few eV, all scattering can be ascribed to the dynamics associated with individual nuclei. This situation corresponds to the so-called neutron Compton scattering or NCS regime. The total $S(\mathbf{q}, \omega)$ can be expressed in term of *intermediate scattering function* and the two-particles correlation function $Y_{jj}(\mathbf{q}, t)$

$$\begin{aligned} S(\mathbf{q}, \omega) &= \frac{1}{2\pi\hbar} \int I_s(\mathbf{q}, t) e^{-i\omega t} dt \\ &= \frac{1}{2\pi\hbar N} \int_{-\infty}^{\infty} \exp(i\omega t) \sum_{jj'} Y_{jj'}(\mathbf{q}, t) dt \end{aligned} \quad (2.53)$$

where N is the number of particles in the target system. The correlation function $Y_{jj'}(\mathbf{q}, t)$ is given by:

$$Y_{jj'}(\mathbf{q}, t) = \langle e^{-i\mathbf{q}\cdot\mathbf{R}_j} e^{i\mathbf{q}\cdot\mathbf{R}_{j'}(t)} \rangle \quad (2.54)$$

where $\mathbf{R}_j(t)$ is the Heisenberg operator for the position of particle j at time t , and $\mathbf{R}_j = \mathbf{R}_j(0)$. We can simplify this description considering the DINS process. This condition amount to considering large $\hbar\mathbf{q}$ and $\hbar\omega$ relative to other momentum and energy scale in the system under investigation. In this case, the neutron-scattering process is adequately described within the framework of the impulse approximation (IA). In the approximation if there are negligible external forces acting on the neutron-nucleus system, the use of momentum and energy conservation laws provides a direct relationship between $\hbar\mathbf{q}$ and $\hbar\omega$ of

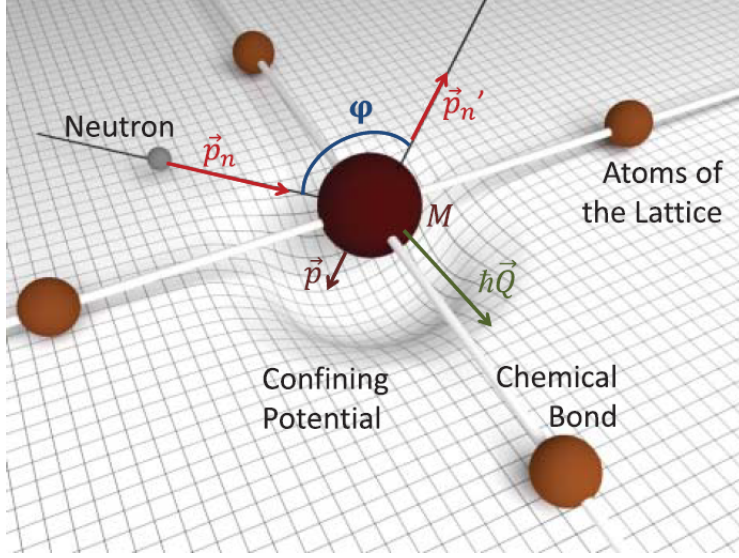


Figure 2.4: Kinematic variables used in the main text to describe the scattering of eV neutrons in a condensed matter system. The curved grid represents the confining potential around a given atom in a lattice. Image courtesy of Ref [12]

the form:

$$\hbar\omega = \frac{(\mathbf{p}_n - \mathbf{p}_n')^2}{2M} + \frac{(\mathbf{p}_n - \mathbf{p}_n') \cdot \mathbf{p}}{M} = \frac{\hbar^2 q^2}{2M} + \frac{\mathbf{q} \cdot \hat{\mathbf{p}}}{M} \quad (2.55)$$

where $\hbar\mathbf{q} = \mathbf{p}_n - \mathbf{p}_n'$, with \mathbf{p}_n and \mathbf{p}_n' are the momenta of the incident and scattered neutron. M and \mathbf{p} are the mass and the momentum of the struck particle before the collision. The scattering of neutrons by single atom, using the high \mathbf{q} values accessible inside DINS experiments, give access to the momentum distribution. The energy distribution of the scattered neutrons, indeed, is directly related to the distribution of particle momenta projected onto the scattering vector \mathbf{q} . For small t , $\mathbf{R}_j(t)$ can be replaced by

$$\mathbf{R}_j(t) \simeq \mathbf{r}_j + \frac{t}{M} \mathbf{p}_j \quad (2.56)$$

where \mathbf{p}_j is the momentum operator of particle j of mass M_j . From this approximation one obtains the correlation function simplify because different nuclei contribution cancel out and the incoherent approximation holds such that only $j = j'$ terms associated with the properties of the *same* particle will be retained. One then obtain:

$$Y_{jj}(\mathbf{q}, t) = \exp \left[\frac{i\hbar t q^2}{2M_j} \right] \left\langle \exp \left[\frac{it}{M_j} \mathbf{q} \cdot \mathbf{p}_j \right] \right\rangle \quad (2.57)$$

The IA approximation to the incoherent dynamic structure factor for an isotropic system is obtained by using equation (2.57) in (2.53):

$$S_{IA}(\mathbf{q}, \omega) = \frac{1}{2\pi\hbar N} \sum_j \int_{-\infty}^{\infty} \exp \left[-i\omega t + \frac{i\hbar t q^2}{2M_j} \right] \left\langle \exp \left[\frac{it}{M_j} \mathbf{q} \cdot \mathbf{p}_j \right] \right\rangle \quad (2.58)$$

Integrating over t :

$$S_{IA}(\mathbf{q}, \omega) = \int n(\mathbf{p}) \delta \left(\hbar\omega - \hbar\omega_r - \frac{\mathbf{p} \cdot \hbar\mathbf{q}}{M} \right) d\mathbf{p} \quad (2.59)$$

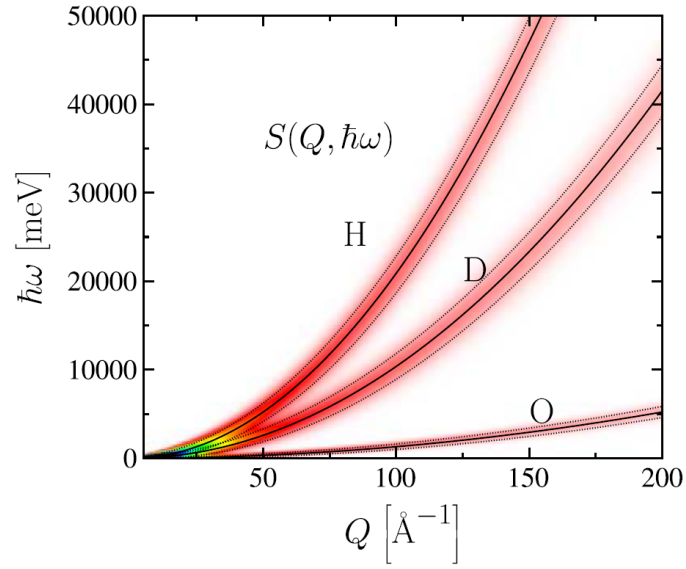


Figure 2.5: Atomic recoil trajectories as a function of momentum and energy transfer (abscissa and ordinate, respectively). The black solid lines denote parabolic recoil lines for several atomic species given by $\hbar\omega = \hbar^2 \mathbf{q}^2 / 2M$. The dashed lines represent the curves $\hbar\omega = \hbar^2 q^2 / 2M \pm \hbar^2 \sigma q / M$, where σ is 5, 7, and 12 \AA^{-1} for H, D, and O respectively. These dashed lines and the shading underneath are used to depict the width and associated extent of the scattering response around the atomic recoil lines. These broadenings are directly related to the overall width of the underlying NMDs, as described in the main text. Image courtesy of [12]

The dynamic structure factor in IA contains the *recoil energy* ($\hbar\omega_r = \hbar^2 q^2/2M$) that is the energy that a stationary particle would have gained from a collision with a neutron owing to momentum and energy conservation. In figure 2.5 we can see the parabolic center ($\omega_r \propto q^2/M$) of dynamic structure factor for different masses.

2.4 Y Scaling and Neutron Compton Profile

In this section we turn to the problem of extracting information about the momentum distribution from measured data. If the conditions for the impulse approximation assumption are fulfilled (high energy and q vectors transfers) the $S(\mathbf{q}, \omega)$ is given by

$$S_{IA}(\mathbf{q}, \omega) = \int n(\mathbf{p}) \delta\left(\hbar\omega - \hbar\omega_r - \frac{\mathbf{p} \cdot \hbar\mathbf{q}}{M}\right) d\mathbf{p} \quad (2.60)$$

One can define now $J(y) = \hbar^2 q/M S(\mathbf{q}, \omega)$ and rewrite the delta function to give:

$$J(y) = \int n(\mathbf{p}) \delta(\mathbf{p} \cdot \hat{\mathbf{q}} - y) d\mathbf{p} \quad (2.61)$$

using the y -scaling by West:

$$y = \left(\frac{M}{\hbar q}\right) \cdot (\omega - \omega_r) \quad (2.62)$$

The form of $S(\mathbf{q}, \omega)$ has an important consequence. Consider an isotropic system, where the scattering is independent of the direction of \mathbf{q} . Then J depends only on the variable y , rather than on \mathbf{q} and ω independently. We are collapsing into a function $J(y)$ of a single variable. This phenomenon is known as *y-scaling*[13], and $J(y)$ is the Compton profile. From the relation 2.61 we can see that the y is the projection of the atom's momentum onto the scattering vector.

The function $J(y)$ is a convenient and standard form for experimental results to be presented. The recoil peak is shifted to be centred at $y = 0$, and if the profile is normalized to unity the mean kinetic energy per atom is directly related to the second moment of the Compton profile by:

$$E_K = \frac{3\hbar}{2M} \int_{-\infty}^{\infty} y^2 J(y) dy \quad (2.63)$$

Chapter 3

The VESUVIO spectrometer

In this chapter we are going to describe the VESUVIO spectrometer, for the modeling and the simulation of the neutronic response of its components is the ultimate goal of this Thesis.

DINS experiments on the VESUVIO spectrometer focus on energy ranges around 1-2 orders of magnitude greater than other thermal neutron instruments (1-150 eV), thus measuring the atomic momentum distribution in condensed matter systems [16], (see Introduction). The reader should recall that at sufficiently high momentum and energy transfers we can use the impulse approximation, implying that the neutron scatters from a single atom, with conservation of the total kinetic energy and momentum of system including the neutron and atom. A photograph of the VESUVIO instrument is shown in figure 3.2 and its schematic representation is presented in figure 3.3. At forward scattering there are 64 (S135–S198 in figure 3.3) cerium doped yttrium aluminum perovskite (YAP) γ -ray detectors. These are at a distance between 50 and 75 cm from the sample position and are arranged in eight columns each containing 8 detectors. Each detector element is 8 cm in height, 2.5 in width and of thickness ~ 0.6 cm. The forward scattering detectors cover a range of angles



Figure 3.1: The VESUVIO Spectrometer at the ISIS Pulsed Neutron and Muon Source

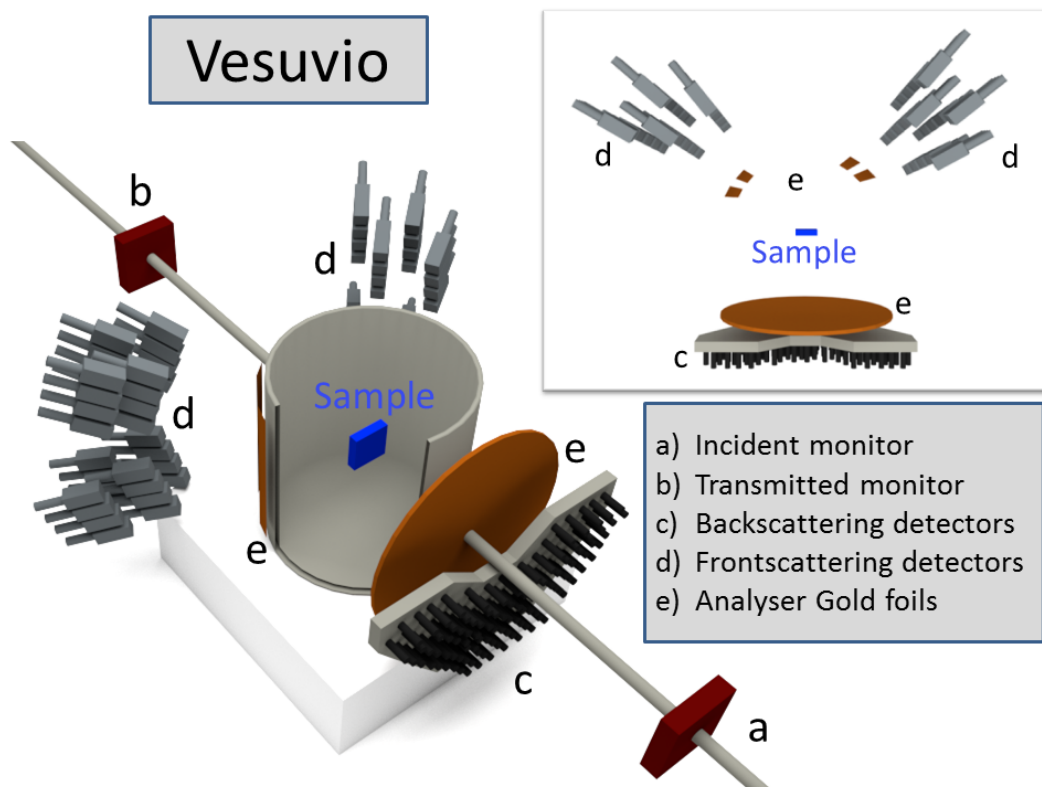


Figure 3.2: The VESUVIO spectrometer at the ISIS Pulsed Neutron and Muon Source (Didcot): The large vertical cylindrical 'sample tank' allows cryostats, furnaces, etc, containing the sample to be inserted in the neutron beam. Neutrons not scattered by the sample pass down the horizontal 'get lost tube' exiting at the bottom of the photograph. The black tubes are photomultiplier tubes which are attached to the YAP γ -ray detectors at forward scattering angles. The FC device used in the FC method [14] at forward scattering angles can be seen attached on each side of the sample tank. The energy selection at back-scattering is performed by the DD technique [15] using the circular disc behind the sample tank. This rotates to three positions, corresponding to no foil, thin foil and thick foil between sample and back-scattering detectors.

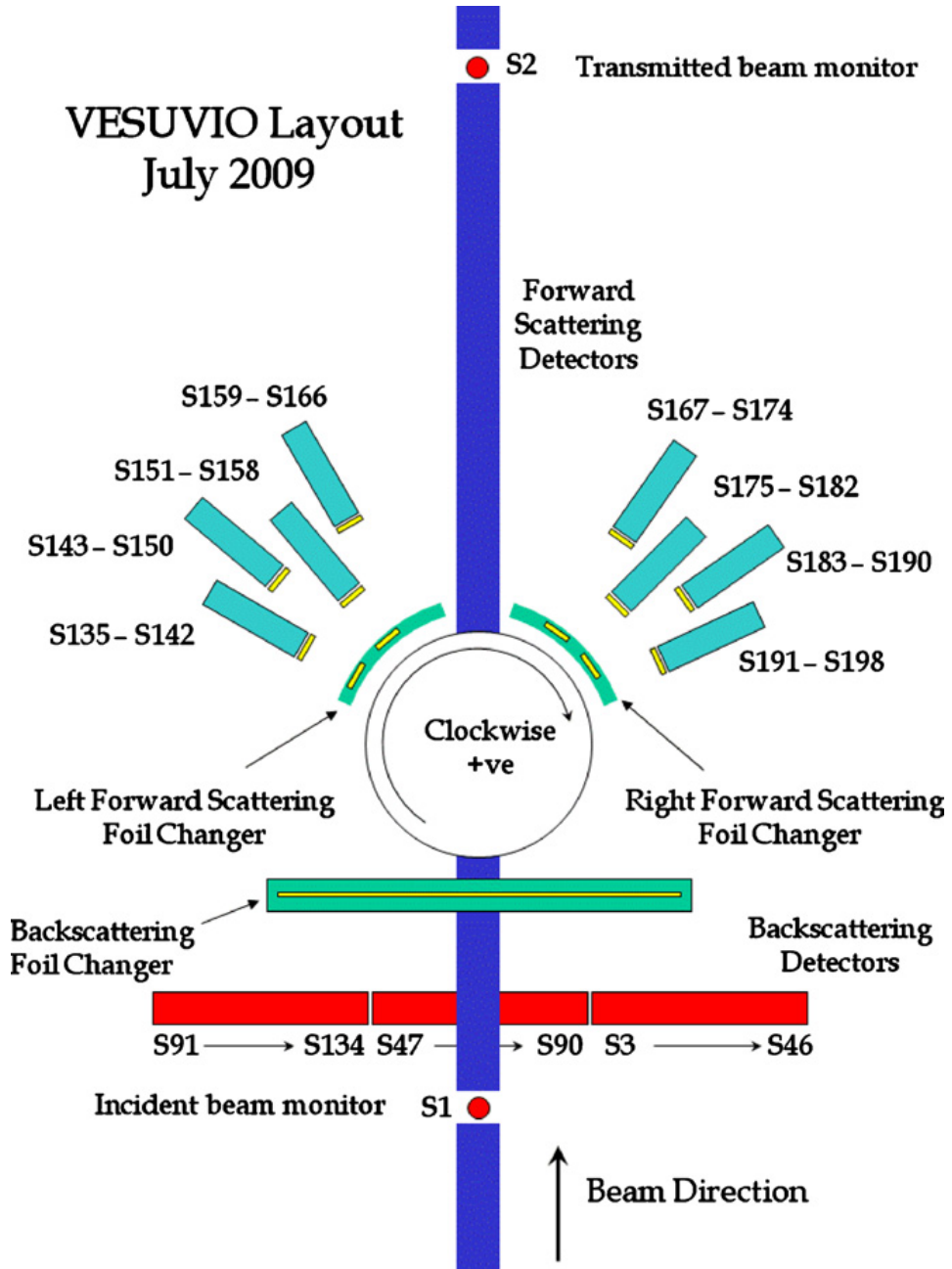


Figure 3.3: Schematic diagram of the VESUVIO instrument. The spectrum numbers corresponding to the different detectors are shown. The positions of the foils used in the differencing measurements are also indicated in yellow. With the foil positions shown the foil is ‘in’ for detectors S135–S142, S151–S158, S167–S174, S183–S190 and ‘out’ for other detectors. By moving the foils sideways the ‘foil in’ detectors become ‘foil out’ and vice versa. In backscattering the detector are numbered S3–S134.

33° – 67°.

At back-scattering there are 132 ^6Li doped neutron detectors (S3-S134 in figure 3.3) arranged in a semi-circle below the scattering plane. These are between 45 and 70 cm from the sample position and have dimensions $4 \times 2 \times 0.6 \text{ cm}^3$. The back-scattering detectors cover a range of angles 130° – 163°.

Gold foils are used to define the energy E_1 of detected neutrons. Gold exhibits an intense neutron-nucleus resonance B.1 peak which strongly absorbs neutron with energies in the range $E_1 \sim 4.9 \pm 0.15 \text{ eV}$.

At forward scattering angles gamma ray detectors are used in the *foil cycling* (FC) technique [14], to define E_1 . At back-scattering neutron detectors with the *double difference* (DD) technique are used. In the latter case a circular disc is rotated every 5 min to one of three positions with no foil, thin foil and thick foil between sample and neutron detector.

3.1 Foil Cycling Technique

The foil cycling technique makes use of the resonant filter ^{197}Au , while ^{238}U was used in the past. The energy-dependent transmission function of a resonant absorbing filter is given by:

$$T(E) = \exp[-\rho\sigma_{eff}x\bar{f}(E)] \quad (3.1)$$

where $\bar{f} = f(E)/f(E_0)$ is the normalized nuclear absorption profile, σ_{eff} the effective absorption cross-section at the peak of the resonance, x the thickness and ρ the nuclear density of the filter, respectively. For the ^{197}Au and ^{238}U filters the Breit-Wigner formula can be approximated by a Lorentzian line shape:

$$L(E) = \frac{1}{\pi} \frac{\Gamma_0}{\Gamma^2 + (E - E_0)^2} \quad (3.2)$$

where E_0 is the resonance energy and Γ_0 is the half-width at half-maximum (HWHM). We remind that this assumption is based on the value Γ_0/E_0 being small, in our case for ^{197}Au and ^{238}U the ratio is lower than 0.02 [15]. As far as the Gaussian line shape (accounting for the Doppler broadening) is concerned [15], it can be shown that its standard deviation, σ_T , is proportional to the square root of the analyzer effective temperature, T^* , times the neutron energy E , i.e.,

$$\sigma_T = \left(2 \frac{m}{M_f} ET^*\right) \quad (3.3)$$

where M_f is the nuclear mass of the material composing the filter and m the neutron mass. The quantity T^* is approximately related to the Debye temperature of the filter, Θ_D , and

to its temperature, T , through the formula [15]

$$T^* \simeq \frac{3}{8} \Theta_D \coth \left(\frac{3}{8} \frac{\Theta_D}{T} \right) \quad (3.4)$$

In a typical VESUVIO scattering experiment, the energy interval of interest is much less than 1 eV around the resonance energy E_0 (with E_0 4.906 eV for ^{197}Au). Within this energy interval, the σ_T dependence on the energy can be neglected. Therefore, as a convolution of the two above line shapes one obtains a Voigt profile:

$$f(E) = \frac{1}{\sqrt{2\pi}\sigma_T} \Re \left\{ \text{erfc} \left(\frac{\Gamma_0 - iE + iE_0}{\sqrt{2}\sigma_T} \right) \exp \left[\frac{(\Gamma_0 - iE + iE_0)^2}{2\sigma_T^2} \right] \right\} \quad (3.5)$$

$$f(E_0) = \frac{1}{\sqrt{2\pi}\sigma_T} \text{erfc} \left(\frac{\Gamma_0}{\sqrt{2}\sigma_T} \right) \exp \left(\frac{\Gamma_0^2}{2\sigma_T^2} \right) \equiv \frac{\psi_0}{\pi\Gamma_0} \quad (3.6)$$

In Eq. 3.6 the coefficient ψ_0 defines the ratio of Voigt profile to the corresponding Lorentzian profile peak at $E = E_0$. When $\sigma_T \rightarrow 0$ (i.e., in the limit $T^* \rightarrow 0$), then $\psi_0 = 1$, the line shape reduces to a simple Lorentzian:

$$f(E) = L(E) \quad (3.7)$$

$$f(E_0) = \frac{1}{\pi\Gamma_0} \quad (3.8)$$

and $\sigma_{eff} = \sigma_0\psi_0$ becomes σ_0 , that is the peak absorption cross-section of a bound nucleus. One should notice that σ_T never reaches a vanishing value, not event at $T \rightarrow 0$, because of the quantum nature of the target nuclei and their zero-point kinetic energies. Defining the dimensionless parameter $\tau = \rho\sigma_{eff}x$, Eq. 3.1 can be rewritten as

$$T(E) = \exp[-\tau\bar{f}(E)] \quad (3.9)$$

and then the single difference transfer function (SD) of the spectrometer can be expressed as

$$X_{SD}(E) = 1 - T(E) \quad (3.10)$$

3.2 Double Difference Method

In the double difference (DD) method one makes use of two filters of thickness x_1 and x_2 , with $x_2 = x_1\beta^{-1}$ ($0 \leq \beta \leq 1$) and the instrument transfer function is the result of the following linear combination:

$$\begin{aligned} X_{DD}(E) &= X_{SD}^{thin}(E) - \beta X_{SD}^{thick}(E) \\ &= 1 - \exp[-\tau\bar{f}(E)] - \beta \{1 - \exp[-\tau\bar{f}(E)/\beta]\} \end{aligned} \quad (3.11)$$

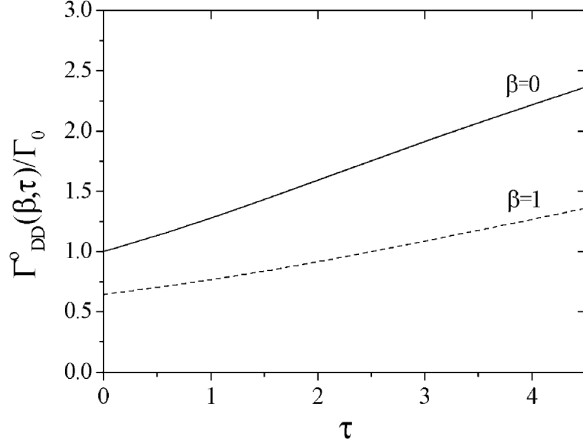


Figure 3.4: Ratio between the HWHM of the DD transfer function in the zero temperature limit, $\Gamma_{DD}^0(\beta, \tau)$ and the Breit-Wigner HWHM, Γ_0 , for both $\beta = 0$ (full line) and $\beta = 1$ (dashed line).

For $\beta = 0$ one obtains $X_{DD}(E) = X_{SD}(E)$, while in the opposite limit: $\beta = 1$ (i.e., two filters of identical thickness), $X_{DD}(E)$ vanishes identically. The HWHM for the transfer function $X_{DD}(E)$, namely $\Gamma_{DD}(\beta, \tau)$, can be calculated analytically only for $\beta = 0$ (SD case) and in the zero effective temperature limit ($T^* = 0$), yielding:

$$\begin{aligned} \lim_{T^* \rightarrow 0} \Gamma_{DD}(0, \tau) &= \Gamma_{SD}^0(\tau) \\ &= \Gamma_0 \left\{ \frac{\tau}{\ln \left[\frac{2}{1 + \exp[-\tau]} \right]} - 1 \right\}^{1/2} \end{aligned} \quad (3.12)$$

where the superscript 0 marks the absence of any Doppler broadening. The growth of $\Gamma_{SD}^0(\tau)$ with increasing τ is shown by the full line in figure 3.4. At the actual configuration the most efficient configuration (see figure 3.5) is obtained using $\beta = 0.28$. In the double difference the foil thick thickness is 50 μm and the foil thin thickness is 14 μm .

3.3 Data Analysis

A single VESUVIO measurement consists of six different configurations of the machine referred to as periods and corresponding to the 3x2 different combinations of FC and DD configurations. When the shutter opens and allows the neutron beam to flow to the sample, the charge of the protons that hit the neutron source is measured by ISIS. The data is collected by detectors for the time necessary to reach 15 μAh and then a new period begins. Every period the back scattering foil changer moves so for every detector we take two measure without foil, two with the thin foil and two with the thick foil. Every periods

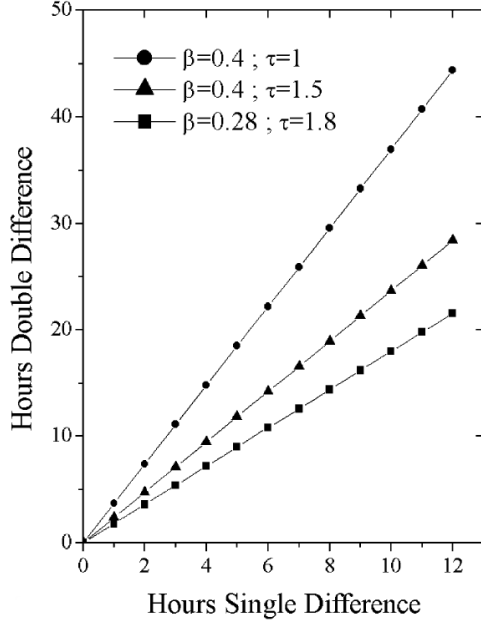


Figure 3.5: Relationships between the time required in DD and SD experiments for the same statistical accuracy for three couples of τ and β values

the front scattering foil changer moves. This foil changer has foils into the scattered beam for half of the detectors every period, and the alternating movement simply changes the detectors filtered by the foil. At the end of the six periods every front scattering detector has measured 3 periods with the 'foil in' configuration and 3 periods with the 'foil out' configuration. At the end of the run the total proton charge flown in the ISIS synchrotron is 90 μAh and the measure ends. With a proton current of 160 μA , the typical flux of the Target Station 1 where VESUVIO is installed, every period takes around 5-6 minutes to complete and a complete run takes around 35 minutes. For the data reduction process the procedure is slightly different for back and front scattering detectors. Nevertheless only the first steps are different and for every detector we reach the I_{out} using the relations X_{SD} for front scattering and X_{DD} for back scattering. From this point the data reduction is the same. Every time the neutron are emitted from the target station a time t_0 is provided by ISIS, a proton beam hits the tungsten target with a frequency of 50 Hz.

In figure 3.6 we show 3 periods for the back scattering detectors. Summing the same configuration periods (1 + 2, 3 + 4, 5 + 6) we obtain the single run intensity for the *thick foil*, *thin foil* and *no foil* configuration respectively. Applying the relation for X_{DD} we obtain the double difference result showed in figure 3.7. For front scattering detectors the same signal is obtained by single difference since the YAP detects the γ directly emitted by the gold resonance. The single difference method is used to subtract the γ background and enhance the resolution. Following this preliminary reduction, spectra relating the physico-chemical

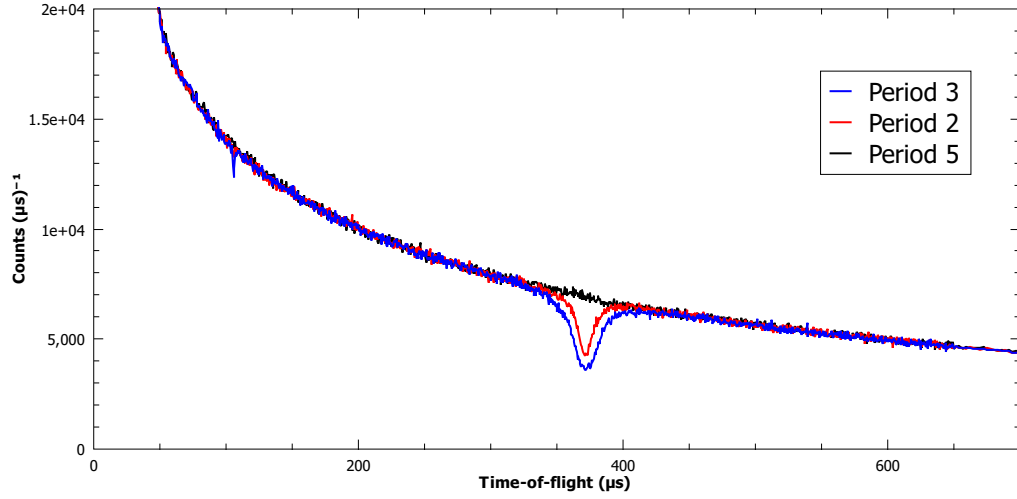


Figure 3.6: Raw data from detector S44 on the VESUVIO spectrometer. The periods define the foils used in the measure: Period 2 Thin foil, Period 3 thick foil and period 5 no foil.

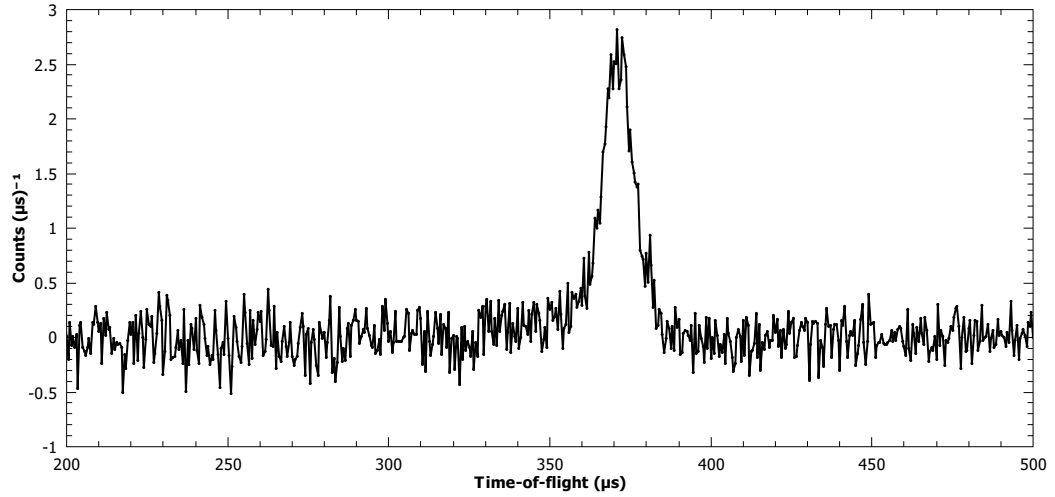


Figure 3.7: Double difference result from raw data in figure 3.6. These are result obtained using the same raw source but we used the routine included into the software that normalize the intensity.

properties of the sample to the energy and momentum transfer are inferred from the angular position of the detector and by the time of flight measure. The time of flight t is determined by:

$$t - t_0 = \frac{L_0}{v_0} + \frac{L_1}{v_1} \quad (3.13)$$

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 energy, $v_1 = \sqrt{2E_1/m}$ the velocity of the detected neutrons with energy E_1 and m is the neutron mass. From t we can calculate E_0 , (E_1 is fixed by foils) using:

$$E_0 = \frac{1}{2}mv_0^2 = \frac{1}{2}m \left(\frac{L_0 v_1}{v_1 t - L_1} \right)^2 \quad (3.14)$$

so we obtain the energy transfer $\omega = E_0 - E_1$ and the momentum transfer

$$q = \sqrt{2m \left(E_0 + E_1 - 2\sqrt{E_0 E_1} \cos 2\theta \right)} \quad (3.15)$$

From knowing the masses M of the struck atoms in the sample and q we can calculate the recoil energy by

$$\hbar\omega_r = \frac{\hbar^2 q^2}{2M} \quad (3.16)$$

and obtain the y -scaling by

$$y = \left(\frac{M}{\hbar q} \right) \cdot (\hbar\omega - \hbar\omega_r) \quad (3.17)$$

summing the contribute from all the backscattering and frontscattering detectors we obtain the result in figure 3.8.

The standard expression for the count rate as a function of t on an inverse geometry neutron spectrometer is [17]

$$C(t) = 2 \left(\frac{2}{m} \right)^{1/2} \frac{E_0^{3/2}}{L_0} I(E_0) D(E_1) N \frac{d^2\sigma}{d\Omega dE_1} d\Omega \quad (3.18)$$

A basic assumption of the IA is that for incident neutron wavelengths much less than the inter-atomic spacing, as is the case on VESUVIO, atoms scatter incoherently. In other words the scattered intensity is the sum of intensities from the individual atoms in the sample. It follows that equation 3.18 can be expressed in the form

$$C(t) = 2 \left(\frac{2}{m} \right)^{1/2} \frac{E_0^{3/2}}{L_0} I(E_0) D(E_1) \sum_M N_M \frac{d^2\sigma_M}{d\Omega dE_1} d\Omega \quad (3.19)$$

where N_M is the number of atoms of mass M in the neutron beam and $d^2\sigma_M/d\Omega dE_1$ is the partial differential cross-section for mass M , $I(E_0)$ is the incident flux and $D(E_1)$ the detector efficiency.

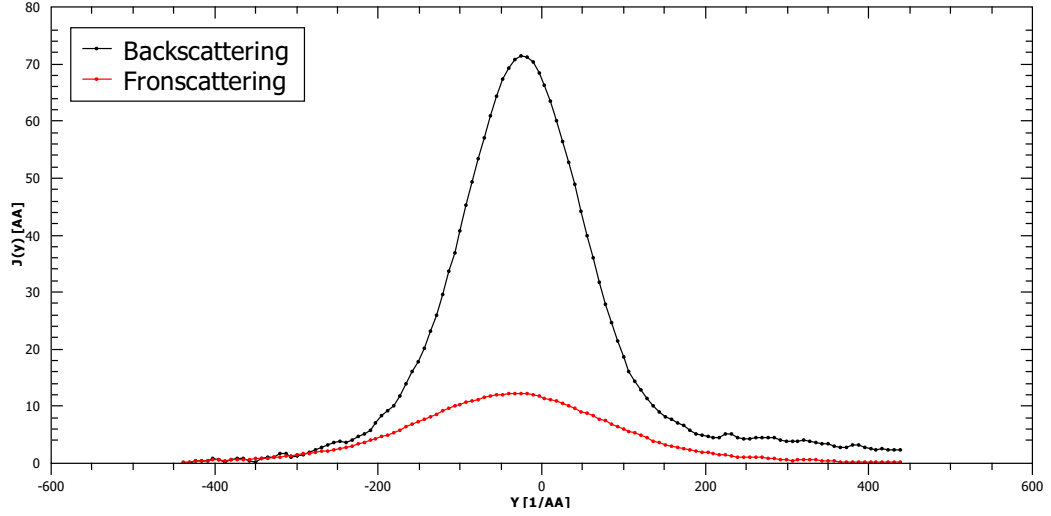


Figure 3.8: Y-scaling from Lead data measured on the VESUVIO spectrometers. The black dots are the sum of backscattering detectors signal and the red dots are the sum of fronscattering detectors

Following the IA formalism with the y-scaling showed in previous chapter we obtain

$$C(t) = \frac{E_0 I(E_0)}{q} \sum_M A_M M J_M(y_M) \quad (3.20)$$

where

$$A_M = \left(\frac{2}{L_0} D(E_R) \sqrt{\frac{2E_R}{m}} \Delta\Omega \right) N_M b_M^2 \quad (3.21)$$

At this point is necessary to define a good function for fitting the data. This task is not trivial because the neutron Compton profile $J(y)$ is not, in general, a simple Gaussian function. It is indeed a convolution of a Lorentzian and Gaussian shape with additional terms taking into account local anisotropy and anharmonicity of the local potential affecting the struck nucleus.

In the framework of the IA, $J(y, \hat{q})$ is the *Radon transform* of the Compton profile $n(\mathbf{p})$ and must be inverted to obtain $n(\mathbf{p})$. The method used to invert $J(y, \hat{q})$ relies upon the following mathematical result [18]. If the $J(y, \hat{q})$ is expressed in the form

$$J(y, \hat{q}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \sum_{n,l,m} \frac{a_{n,l,m}}{2^{2n+l}n!} H_{2n+l}\left(\frac{y}{\sigma\sqrt{2}}\right) Y_{l,m}(\hat{q}) \quad (3.22)$$

then

$$n(\mathbf{p}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{p^2}{2\sigma^2}\right) \sum_{n,l,m} (-1)^n a_{n,l,m} L_n^{l+1/2}\left(\frac{p^2}{2\sigma^2}\right) Y_{l,m}(\hat{p}) \quad (3.23)$$

where H_{2n+l} is a Hermite polynomial, $Y_{l,m}(\hat{q})$ is a Legendre polynomial and $L_n^{l+1/2}(p^2)$ is a Laguerre polynomial. For samples having no preferred orientation, such as liquids or

amorphous solids, there is no dependence on the direction \hat{q} and only the terms $l, m = 0$ contribute. Equations 3.22 and 3.23 reduce to

$$J(y, \hat{q}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y^2}{2\sigma^2}\right) \sum_n \frac{a_n}{2^{2n}n!} H_{2n}\left(\frac{y}{\sigma\sqrt{2}}\right) \quad (3.24)$$

$$n(\mathbf{p}) = \frac{1}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{p^2}{2\sigma^2}\right) \sum_n (-1)^n a_n L_n^{1/2}\left(\frac{p^2}{2\sigma^2}\right) \quad (3.25)$$

Once the peak from a single mass has been isolated in the time of flight spectrum the fit is performed using the 3.20 with σ and a_n as parameters.

3.3.1 Correction to the Impulse Approximation

The Impulse Approximation is exact only in the limit of $q \rightarrow \infty$, in our case, even with a q value of the order of hundreds of \AA^{-1} , it is naturally a finite quantity. We need to take into account the deviation from the ideal IA and correct the data by the *final state effect* (FSE). From the Sears work [19] at finite q , $J(y)$ has the form

$$J(y) = J_{IA}(y) - \frac{M\langle\nabla^2 V\rangle}{36\hbar^2 q} \frac{d^3 J_{IA}(y)}{dy^3} \quad (3.26)$$

where $J_{IA}(y)$ is the impulse approximation result. $\langle\nabla^2 V\rangle$ is the mean value of the Laplacian squared of the potential energy of the struck atom. The expansion has higher order terms which are proportional to $1/q^2, 1/q^3$ and so on. However at the q values on VESUVIO, these are negligible at the current level of statistical accuracy.

Chapter 4

Brief introduction to McStas

The design of new neutron instruments and the upgrade of existing facilities strongly depend upon simulation codes for neutron transport [7, 20]. In recent years the McStas code [1] became one of the most used pieces of software in neutron-scattering simulation. The package is based upon a special meta-language designed for Monte-Carlo ray-tracing calculations, and includes a library of highly-adaptable *components*, with geometrical and physical properties that can be adjusted to the instrument design. The development of these tools was supported with validation with experiments, however, only on a restricted dataset based on the most common kind of instruments. In many cases, neutron instruments are designed so as to operate in a narrow range of energies, as for example diffractometers using cold neutrons and traditional spectrometers using thermal neutrons. As the operational energies of the instrument are fixed, the modeling of the incident and scattered neutron spectra can be verified over limited energy ranges, where reliable experimental data from the real instrument can be provided. ¹ From the first chapter we describe the difficulties to calculate the behavior of a neutron scattering instrument. It can be described in principle by a complex integral over all relevant parameters, like initial neutron energy and divergence, scattering vector and position in the sample, etc.. However, in most relevant cases, these integrals are not solvable analytically and we hence turn to Monte Carlo methods. A totally realistic semi-classical simulation will require that each neutron is at any time either present or lost. In many instruments, only a very small fraction of the initial neutrons will ever be detected, and simulations of this kind will therefore waste much time in dealing with neutrons that never hit the relevant detector or monitor. An important way of speeding up calculations is to introduce a neutron *weight factor* for each simulated neutron ray and to adjust this weight

¹This section takes information from the McStas manual

according to the path of the ray. For example if the reflectivity of a certain optical component is 10%, and only reflected neutrons-ray are considered later in the simulations, the neutron weight will be multiplied by 0.10 when passing this component, but every neutron is allowed to reject in the component. In contrast, the totally realistic simulation of the component would require in average ten incoming neutrons for each rejected one. Let the initial neutron weight be p_0 and let us denote the weight multiplication factor in the j 'th component by π_j . The resulting weight factor for the neutron ray after passage of the n components in the instrument becomes the product of all contributions

$$p = p_n = p_0 \prod_{j=1}^n \pi_j \quad (4.1)$$

Each adjustment factor should be $0 < \pi_j < 1$, except in special circumstances, so that total flux can only decrease through the simulation. Performing a simulation using McStas can be divided into the following steps/elements (refer to figure 4.1 to see an overview of the logic inside the McStas environment):

- To use McStas, an instrument definition file describing the instrument to be simulated must be written.
- Next, the McStas compiler `mcstas` is invoked to translate the instrument and component files into a C program. The program `mcstas` itself is written in C, using the parser *flex* and the compiler *bison*.
- The resulting C program can then be compiled with a C compiler and run in combination with various front-end program for example to present the intensity at the detector as a motor position is varied.
- The output data may be analyzed and visualized in the same way as regular experiments by using the data handling and visualization tools in McStas based on Perl/Python in combination with `chaco`, `matplotlib`, Matlab, GNUPlot or PG-PLOT. Further data output formats including NeXus are available

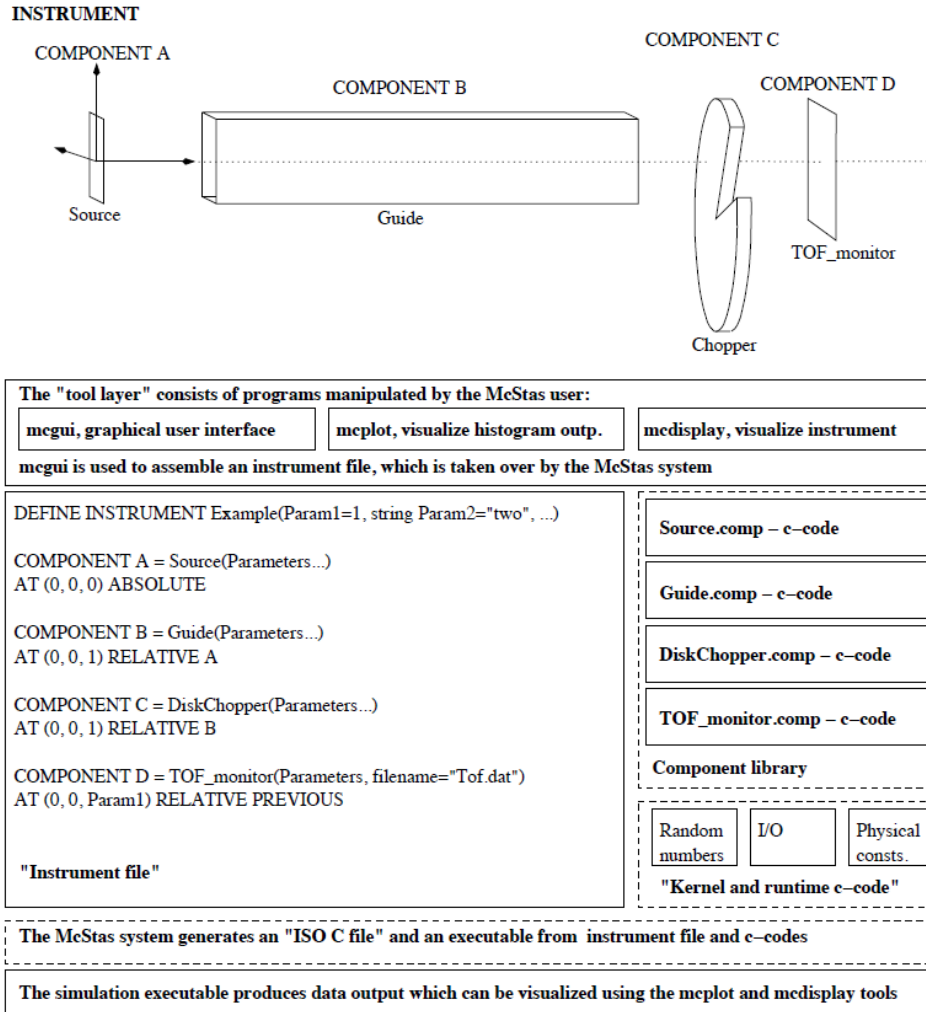


Figure 4.1: An illustration of the structure of McStas.

4.1 The instrument file

The instrument file is the core of the McStas simulation. It describes entirely the source, the elements of the instrument and the detector output using a modular formalism. As a summary, the usual grammar for instrument descriptions is

```

1 DEFINE INSTRUMENT name ( parameters )
2 DECLARE
3   C_code
4 INITIALIZE
5   C_code
6 TRACE
7   components
  
```

```

8 {FINALLY C code}
9 END

```

From the instrument definition head we can pass parameters to the simulation, allowing to adapt the same instrument to the user necessities.

The **DECLARE** section is used for variable declaration. These variables have as scope the entire simulation, so we can see them as global variables. The neutron parameters (**x,y,z,vx,vy,vz,sx,sy,sz,t**) are defined inside the McStas compiler so we do not need to declare them. They are therefore global variables as well, so we must avoid conflicts in declarations.

The **INITIALIZE** section executes the inner C code when the simulation starts, variables are not updated for each iteration.

The **TRACE** section is where the **COMPONENTS** are inserted. The usual grammar for component instances within this section is

```

1 COMPONENT name = comp(parameters)
2 AT (...) [RELATIVE [reference | PREVIOUS] | ABSOLUTE]
3 ROTATED [RELATIVE [reference | PREVIOUS] | ABSOLUTE]

```

The **comp** value is the link to the component used (a source, a sample or an optics element). The position flags like **AT** and **ROTATED** need to be associated to a coordinate reference system. The **ABSOLUTE** is the default reference system, the **PREVIOUS** set the origins at the position of the previous component or we can refer to a specific component inserting the **name**.

We conclude here this introduction: there are other flags and sections in McStas but we are going to explain only the functions implemented in this work.

Part II

The VESUVIO spectrometer in McStas

Chapter 5

Collimation

In this scenario, the VESUVIO spectrometer [21, 22] at the ISIS pulsed neutron and muon source [23], represents a challenging instrument to be modelled. Owing to the absence of permanent energy filters on both the primary and the secondary paths, VESUVIO allows for concurrent measurements over a broad energy range from the fraction of meV to tens of keV. In recent years, VESUVIO has been referred to as an *epithermal and thermal neutron analysis station* [12, 24], as the same sample can be investigated using deep inelastic neutron scattering [25, 26], neutron diffraction [27, 28], neutron transmission [29, 30], and gamma dopplerimetry [24, 31].

This section introduces the original work carried out in this thesis. The tasks under my responsibility consisted in producing a realistic McStas model of VESUVIO, suitable for implementing detailed optimizations and validation of instrumental performance and comparison with experiments. In particular we split the modelling process into three part:

- Collimation: Moderator and primary flight path components analysis;
- Sample Interaction: Testing existing sample `components` and write a new dedicated one;
- Gold Resonance: Write down an analyser foil transfer function component.

In this first part, we present some preliminary comparisons of the simulated incident neutron spectrum and beam profile against recent experimental characterizations[22] as well as simulations of multiple scattering contribution and gamma background evaluations. Differently from DINSMS code where the input flux is generated by an approximate function [5], McStas uses a custom source that can replicate the measured flux on the instrument.

5.1 McStas Model

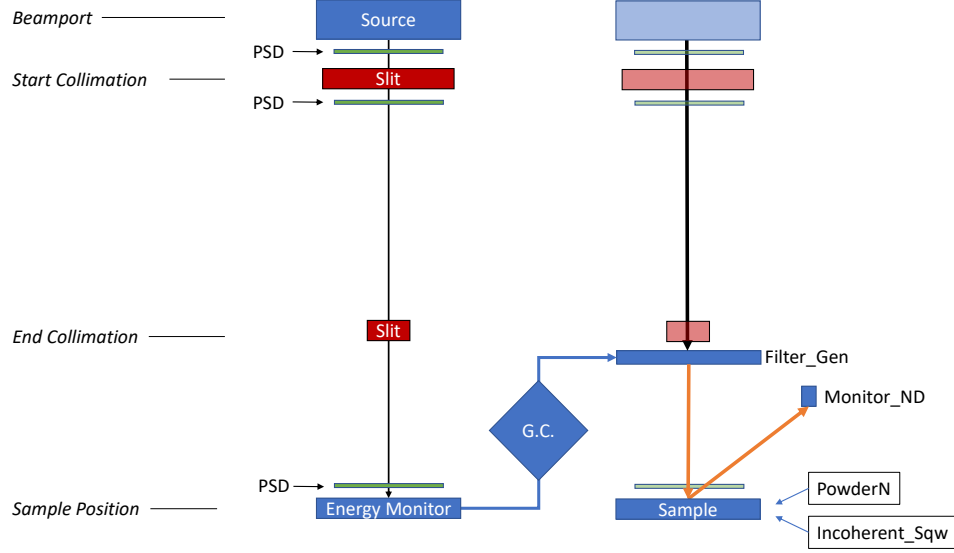


Figure 5.1: Simulation setup for collimation analysis (left) and backscattering analysis (right) after the inclusion of the “generating correction” (GC) component, “Filter_Gen”.

Monte-Carlo calculations have been performed using McStas 2.4.1 [32]. Simulations were run on a local computer with Windows 10 operating system, as well as on the SCARF cluster [33]. The VESUVIO spectrometer, for which a detailed description can be found in Reference [22], was simplified in McStas with a moderator acting as the neutron source, a series of collimation slits along the incident neutron path, a sample at about 11 m from the moderator, and a monitor representing a ^6Li -doped scintillating detector in backscattering, as schematically showed in Figure 5.1. In particular,

- The moderator component used was `Commodus_I` [34] (the most recent version of view-ModISIS [35]), in order to take into consideration the recent upgrade of the water moderator serving the VESUVIO instrument at the ISIS Target Station 1 [22]. This component can simulate the emission of neutrons from the moderator face with a proton current of $1\mu\text{A}$.
- The set of B_4C collimation slits in the range of distances 1.71 m to 9.66 m from the moderator were modelled by ideal `Slit` elements placed at the positions 1.71, 8.46 and 9.66 m. In particular, as the collimation stages are octagonal in transverse cross section, two square slits titled by 45 degrees with respect to each other were used.

- Images of the beam shape and intensity were obtained using virtual Position Sensitive Detector (PSD) components and virtual monitors `PSD_monitor` before and after the first slit, and at the sample position.
- A `Filter_gen` component at the sample position was used to take into account discrepancies between the simulated and experimental beam profiles prior to the scattering from the sample. The `Filter_gen` component reads a table generated from the left setup in figure 5.1, and corrects it to match the experimental one. The corrected flux is then back-traced to the last slit position and used to illuminate the sample. The table is an ASCII file where in the first column are listed the energy values and in the second column are reported correcting factors corresponding to the ratio of experimental-to-simulated intensities, so as to correct for the discrepancies in the two intensities.
- The components used to reproduce the backscattering data from a Pb sample were `PowderN` [36] and `Isotropic_Sqw` [37].
- The simulated spectra corresponding to a backscattering detector were recorded using a `Monitor_nD` component, and the count rate was corrected by the known efficiency of ^6Li -detector efficiency $\eta(E)$ according to

$$\eta(E) = 1 - e^{-n\sigma(E)d}. \quad (5.1)$$

where n is the atom density, $\sigma(E)$ cross section and d detector thickness.

5.2 Collimation results

Inspection of Figure 5.2 shows how the collimation stages modify the shape of the beam from a large square of about 8 cm side to a circle of diameter about 4 cm. The sole reduction of the size of the beam corresponds to a decrease of about one order of magnitude in the beam intensity at the sample position compared to that before the collimation. Indeed, the ratio of simulated beam fluxes at the two positions is found constant, with no visible dependence upon the incident neutron energy. Moreover, the original beam, homogeneous in the (x, y) plane transverse to the sample-moderator direction, is modified into a Gaussian-like profile. The file representing the brightness of the upstream water moderator was provided by the ISIS Neutronics Group and was created using current MCNP-X model of ISIS Target Station 1 [20].

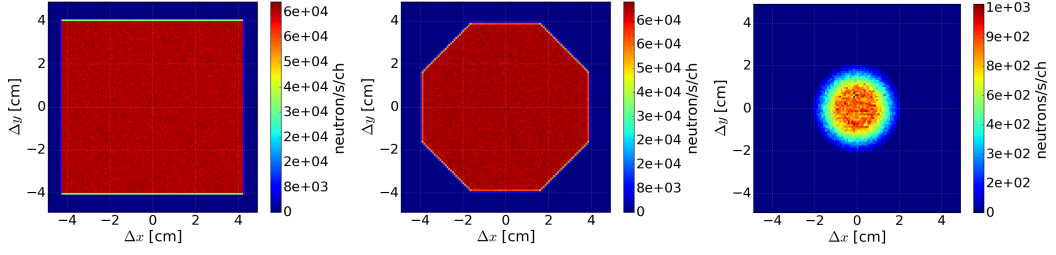


Figure 5.2: Transverse shape of the neutron beam before (left) and after (center) the beam port slit. (right) Shape for the neutron beam at sample position. The PSDs consist on a 120x120 pixel square matrix with 10 cm side length.

The simulated results for the energy spectrum and the beam profile of incident neutrons at the sample position were compared with recent experimental measurements [22]. As the result from the simulation are expressed for unit proton current, the simulated spectra were multiplied by the actual proton current ca. $155 \mu A$ to TS1 during the measurement. The experiment was performed with the detector and about 1.5 m of the beamline in air, causing an attenuation of the neutron beam of about 0.895 with respect to standard measurement in vacuum. Moreover, the neutron beam needs to cross about 1.5 cm of Al windows between the moderator and the detector, therefore requiring an additional attenuation factor of about 0.855. Figure 5.3 shows how the present model gives an absolute intensity of the neutron beam between 1.5 and 3 times larger than the experimental data [22]. The agreement between experiment and simulation improves as the incident neutron energy approaches the epithermal region. Similar discrepancies have been previously discussed in Ref[38] about the modelling of the TOSCA spectrometer, where a ratio of simulated-to-experimental intensities ca. 1.77 was reported between 0.28 \AA and 4.65 \AA , that favorably compares to the factor 2.39 in our case and in the same energy window. In particular, the flux in the epithermal region is expected to behave as $\propto E^{\alpha-1}$ for real moderators with α leakage exponent. By fitting this model to the experimental and simulated data, we obtained $\alpha = 0.04$ and $\alpha = 0.08$ for simulation and experiment, respectively. Such a small discrepancy could be related to the uncertainties present in the MCNP model which the table of emission TS1_S02_Vesuvio.mcstas that describes the moderator in McStas is based on. It is interesting to notice how, despite the discrepancy in the absolute value of the flux intensity, the simulated shape of the beam compares very favourably to the experimental results. The slightly wider experimental profile, shown in Figure 5.4, can be partially explained by the broadening due to the spatial resolution component from the nGEM detector used in Ref. [22].

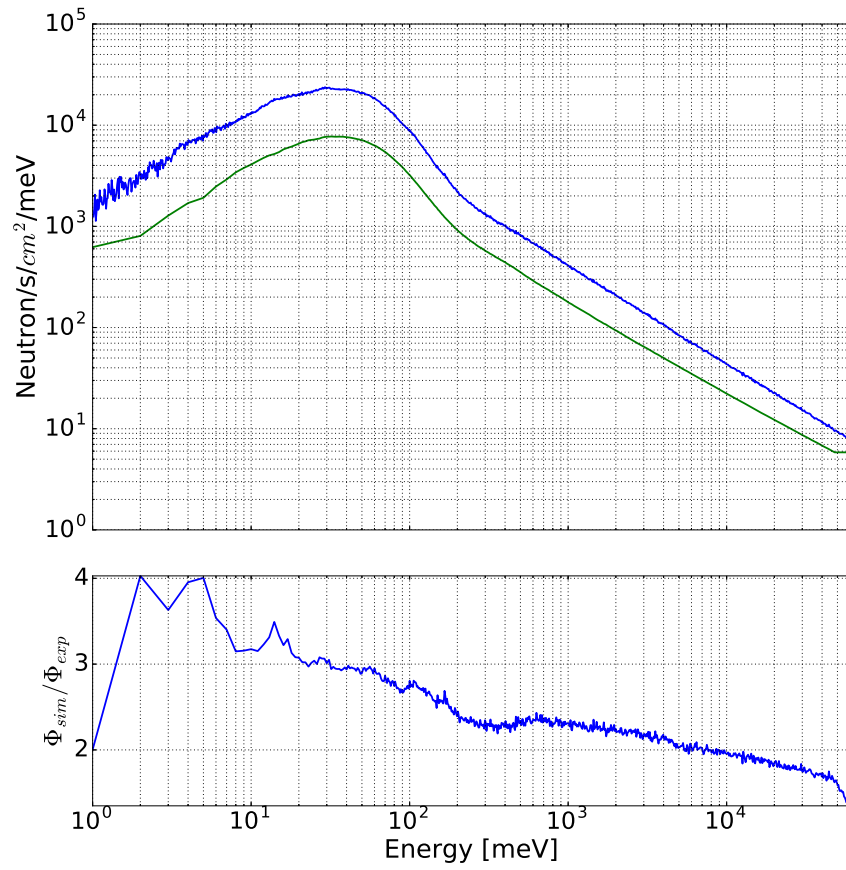


Figure 5.3: Simulated (blue) and experimental (green) fluxes at the sample position, and the simulated-to-experimental ratio in the bottom panel.

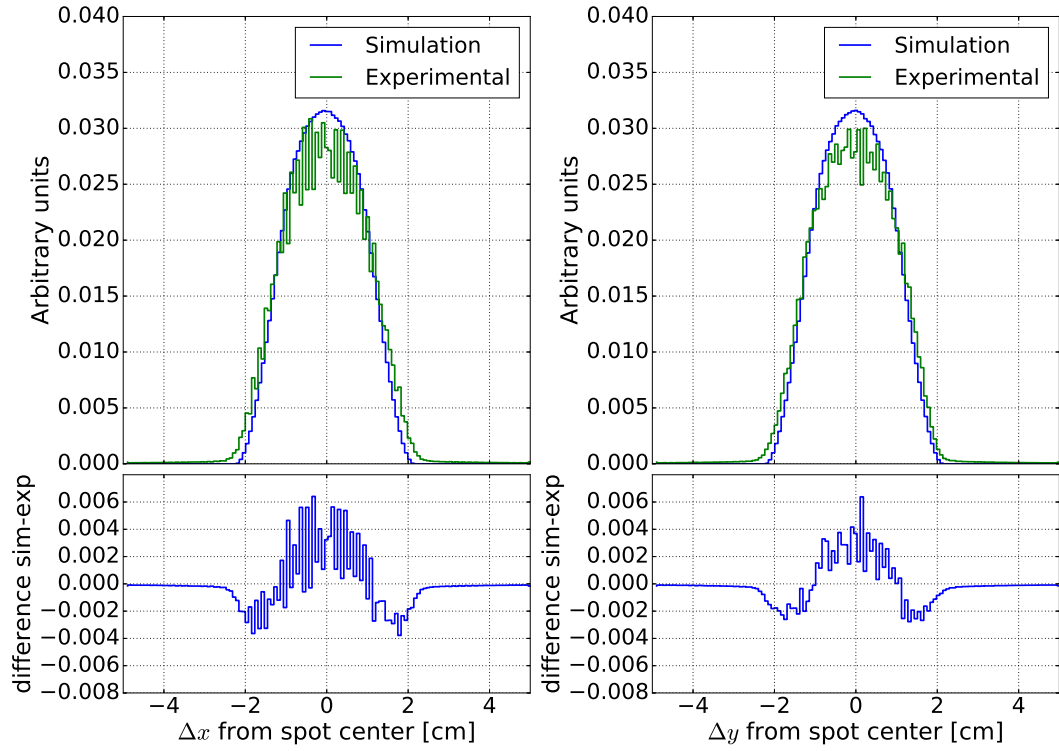


Figure 5.4: Simulated (blue) and experimental (green) along the two directions in the transverse plane, as well as their difference in the bottom panel. The experimental beam profiles from Ref. [22] have been shifted in order to share the same axis as in the simulation. Data is normalized to unit area.

Chapter 6

Sample

Historically, the component development in McStas and its applications did not take advantage of epithermal neutrons available in spallation neutrons sources. However the moderator component provided by the ISIS neutronics group [34] include neutron energies from few meV up to hundreds of keV, and the code itself compiles without problems at any energy range. The main reason is that this first part of the instrument file utilizes only components that modify the weight of the neutrons. If there is no calculation involving energy or velocity rounding errors emerge rarely. There are in any case some limitations carried out by ASCII tables and sample components. The large energy range covered by VESUVIO challenges the functioning of some routines like the `read_table` library. Optimizing the code on a few meV range can give an easy choice to perform linear binning so the library actually rebin linearly every data file. In our case when the energy range start from 0.5 meV to 100 eV the logarithm scale is preferable. The reading of tables and `.dat` files is the most reliable source of *segmentation fault* errors and the rebinning can easily be the reason: for example, in our code two files of 340 lines x 2 columns become enough to fill 10GB ram free-space. We needed to avoid this situation by generating a linear file `.dat`, losing definition.

The other difficulty, but connected to the latter, is the implementation of the specific neutronic response in the eV range in neutrons simulations, as we can see in section 6.2. We have used a `Isotropic_SQW` component where we have uploaded an Impulse Approximation $S(\mathbf{q}, \omega)$ (fig 6.4) generated by a python code, with linear binning.

Before attempting to write our `component` a few simulations with the sample components provided by McStas were performed.

6.1 PowderN

The **PowderN** component can handle mainly the Bragg scattering. It takes as input a data file where information is stored including d-spacing, peaks intensities, directions and so on. The VESUVIO spectrometer can obtain quite good Bragg reflection peaks at backscattering (Figure 6.2), indeed this are used for calibration [39], and interestingly this component can be used as a benchmark of the model geometry and for possible corrections. The assumption for the **PowderN** component is that the orientation of the crystallites is evenly distributed, and there is thus always a large number of crystallites oriented to fulfill the Bragg condition

$$n\lambda = 2d \sin \theta \quad (6.1)$$

where n is the order of the scattering (an integer), λ is the neutron wavelength, d is the lattice spacing of the sample, and 2θ is the scattering angle. As all crystal orientations are realised in a powder sample, neutrons are scattered within a *Debye-Scherrer cone* of opening angle 4θ [40]. We can rewrite equation 6.1 into the form

$$|\mathbf{q}| = 2|\mathbf{k}| \sin \theta \quad (6.2)$$

where \mathbf{q} is a vector of the reciprocal lattice, and \mathbf{k} is the wave vector of the neutron- It is seen that only reciprocal vectors fulfilling $|\mathbf{q}| < 2|\mathbf{k}|$ contribute to the scattering. For a complete treatment of the powder sample one needs to take into account all these \mathbf{q} -values, since each of them contribute to the attenuation. The strength of the Bragg reflections is given by their structure factors

$$\left| \sum_j b_j \exp(\mathbf{R}_j \cdot \mathbf{q}) \right|^2 \quad (6.3)$$

where the sum runs over all atoms in one unit cell. This structure factor is non-zero only when \mathbf{Q} equals a reciprocal lattice vector. The component calculates the scattering cross section summing over essentially different reflections multiplied by their multiplicity, j ¹. Then, a finite packing factor, f , is defined for the powder, and finally, the Debye-Waller factor is multiplied on the elastic cross section to take lattice vibration into account (without inelastic background). In the thin sample approximation the result is

$$\sigma_{cone,q} = f \exp(-2W) \frac{N}{V_0} \frac{4\pi^3}{k^2} \frac{j_q |F(q)|^2}{q} \quad (6.4)$$

For sample of finite thickness, the beam is being attenuated by the attenuation coefficient:

$$\mu_q = \sigma_{cone,q}/V \quad (6.5)$$

¹The textbook expression for the scattering cross section corresponding to one Debye-Scherrer cone is proportional to $\sum_q |F(q)|^2$, where $F(q)$ is the structure factor for a particular q .

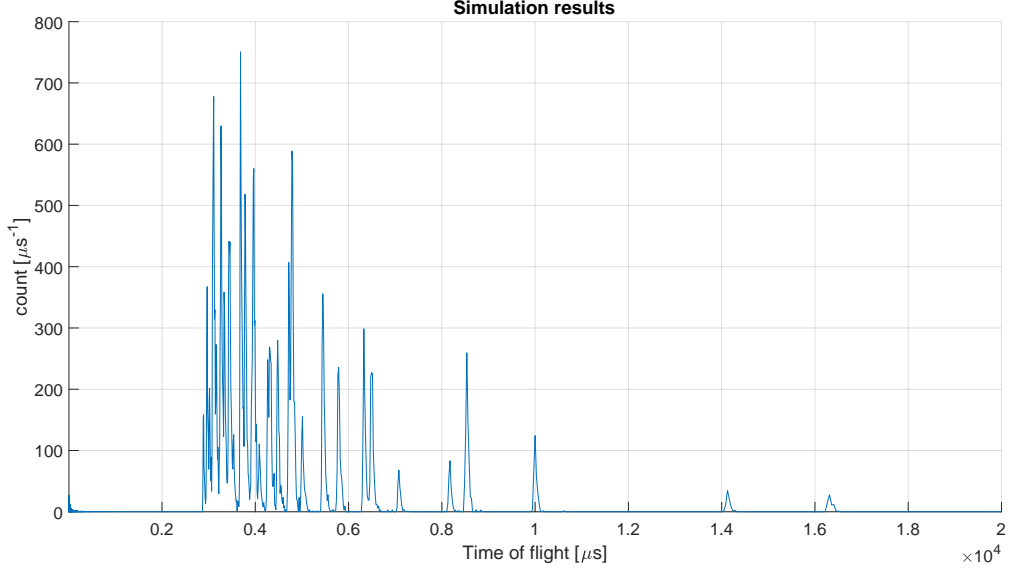


Figure 6.1: Simulation of Bragg spectra from S44 detector.

Using the `PowderN` to simulate a lead sample in McStas on the VESUVIO model, one obtains the result in Figure 6.1. The component can calculate incoherent background but there are remarkable differences from the experimental data shown in figure 6.2. The main reason is that the incoherent background use the cross section at 25 meV without taking into account the inelastic scattering. At the energies above 25 meV, corresponding of TOF below $5 \times 10^4 \mu\text{s}$ the inelastic scattering become predominant but such contribution is not computed.

At least some useful results can be extrapolated by this component. During the modeling of the instrument we used blueprints and experimental calibration together to set detector position or collimation structure dimensions. Moreover in the source component we are using another simulation by MCNPX of the target station that set the t value by a monitor placed to the beam port of VESUVIO beam line, and than backtraced to the moderator face in run-time. Despite the several approximations, the simulated results show discrepancies of less than 0.3% with respect to the experimental data as shown in figure 6.3. By fitting this discrepancy we were able to reconstruct the path difference from experiment to simulation and it is about 2.2 cm. This is probably due to the real distance included in the moderator file. Considering the thickness of the moderator (about 4.5 cm) seems reasonable considering this shift as a consequence of the calibration, that consider the distance as the effective neutron emission position.

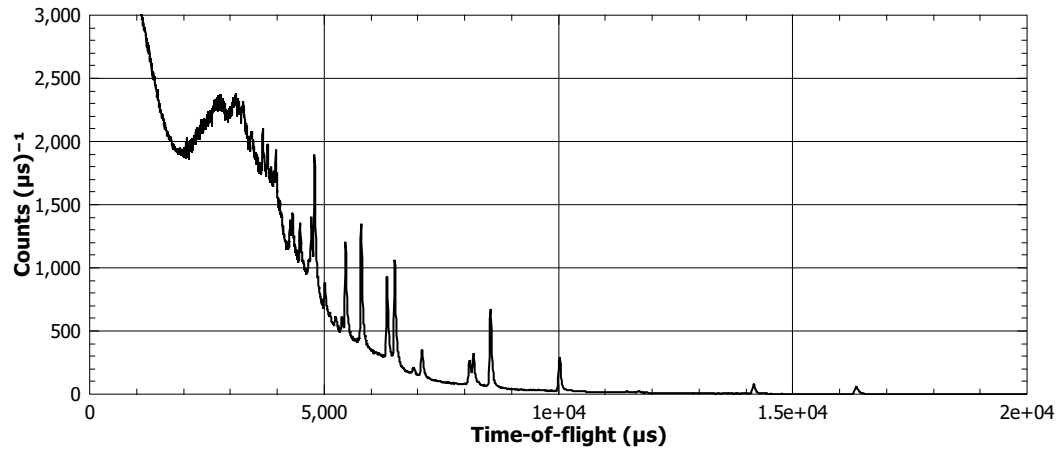


Figure 6.2: Experimental RAW data from the detector S44.

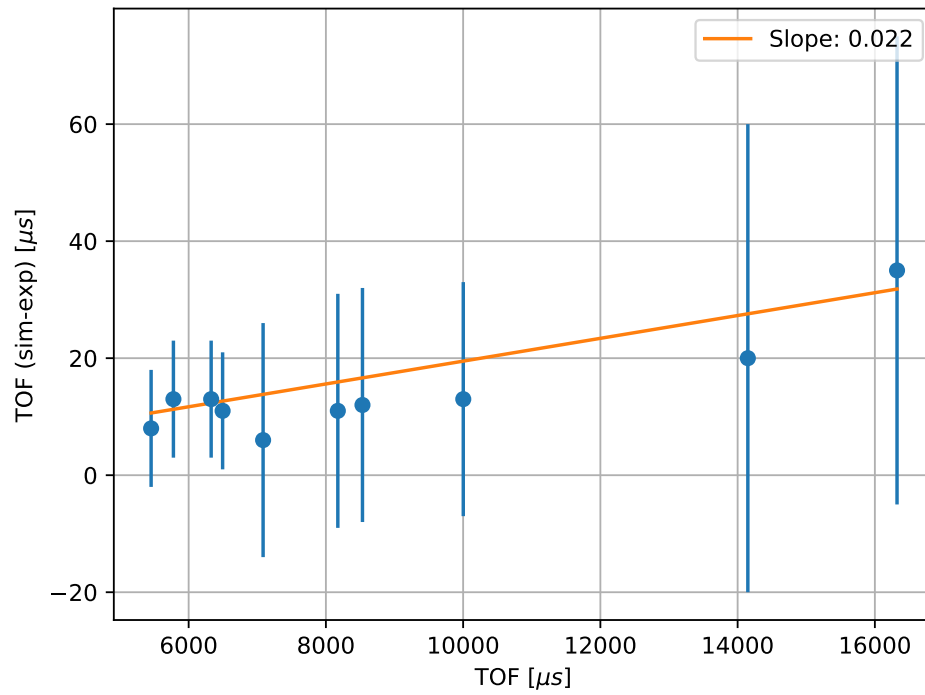


Figure 6.3: Variations in TOF of the simulated Bragg Peaks from experimental data.

6.2 Isotropic_SQW

The sample component `Isotropic_Sqw` has been developed in order to simulate neutron scattering from any isotropic material such as liquids, glasses (amorphous systems), polymers and powders (currently, mono-crystals cannot be handled). The component treats coherent and incoherent neutron scattering and may be used to model most materials, including sample environments with concentric geometries. The structure and dynamics of isotropic samples can be characterized by the dynamic structure factor $S(\mathbf{q}, \omega)$, which determines the interaction between neutrons and the sample and therefore can be used as a probability distribution of ω -energy and q -momentum transfers. It handles coherent and incoherent processes, both for elastic and inelastic interactions. The main input for the component is $S(\mathbf{q}, \omega)$ tables, or powder structure files.

The VESUVIO spectrometer can reach values of the moment transfer $20\text{\AA} < q < 150\text{\AA}^{-1}$ within a energy range of 5 eV to 150 eV. In this region of $S(\mathbf{q}, \omega)$ the scattering is well described, as already mentioned, by the Impulse Approximation and, for this reason, this component is the main candidate to simulate the scattering properties of samples and equipment on the VESUVIO spectrometer. The input table has been generated from the $S(\mathbf{q}, \omega)$ using a simple Python code (Appendix A). For this particular simulation we use only the IA as a coherent input file (if one does not provide two distinct files for coherent and incoherent scattering the component uses the coherent as the total dynamic structure factor). Result is showed in figure 6.6. A particular detail of the computation process inside this component is quite long so we do not cover it in this thesis². Some notable defects on the other way limits a complete use for our scope. The simulation can exploit this component only for heavy-weight elements, such as Pb, or using a small range of neutron energy emitted by the source. For the lead we can lower this range to 3000 meV to 6000 meV, if we need a larger energy range the code crash. The lead however has a really low recoil energy due to his mass of 207 a.m.u., so the scattering is nearly elastic and the neutron does not lose much energy. At the moment we can no set a energy range value from 1 eV to 120 eV, which is needed to cover the impulse approximation for light atoms (such H or D).

²A complete description is available on McStas component guide <http://www.mcstas.org/documentation/manual/>

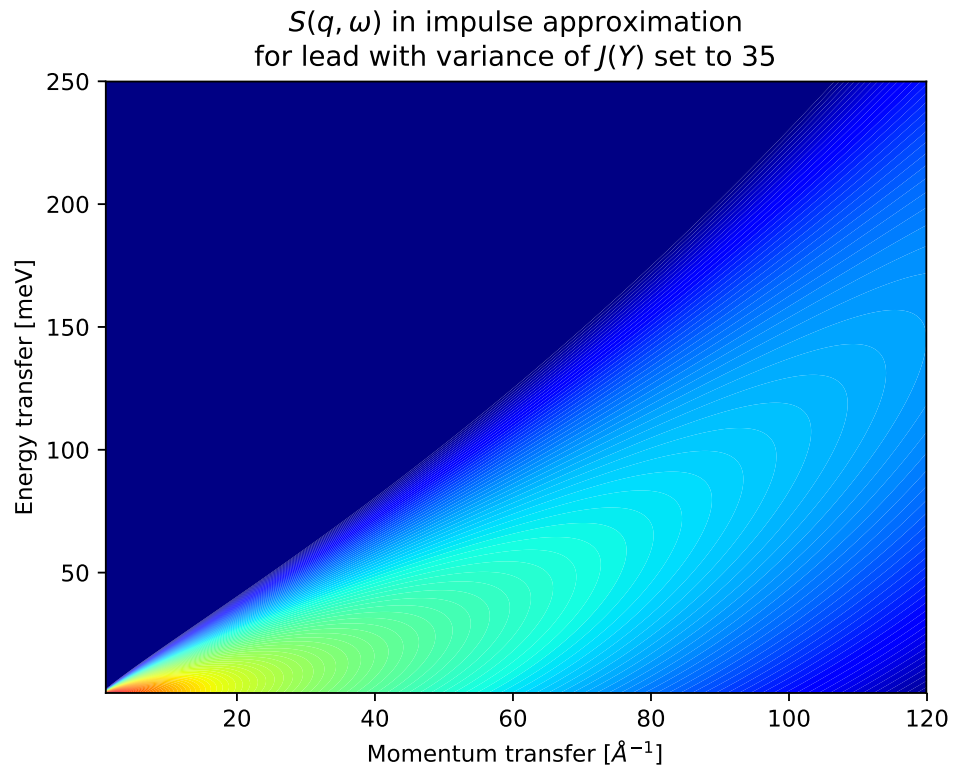


Figure 6.4: Plot of the lead $S(\mathbf{q}, \omega)$ obtained by the algorithm written in Appendix A

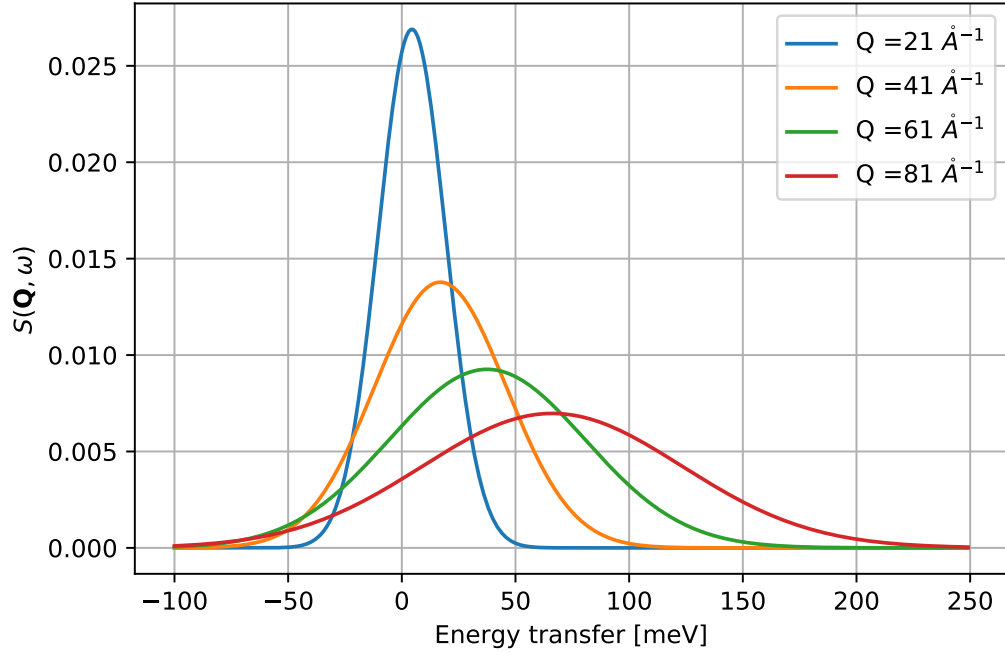


Figure 6.5: Plot of the lead $S(\mathbf{q}, \omega)$ obtained by the algorithm written in Appendix A for a discrete set of energy transfer.

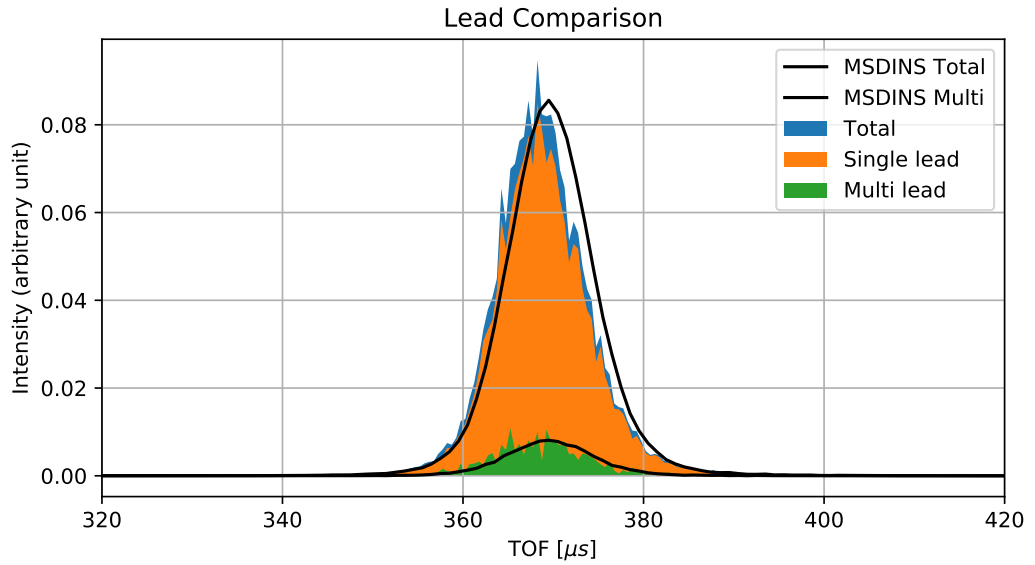


Figure 6.6: Lead simulation using the `Incoherent_SQW`.

Chapter 7

The Impulse Approximation Component

All previous components have been optimized for particular tasks (e.g. reproducing Bragg reflections), or they are more complex and written so as to be as adaptable as possible to one's particular task.

For example the `Incoherent_SQW` takes as input file a user-provided mesh file containing the $S(\mathbf{q}, \omega)$. At present, McStas accepts ASCII files of the form ω_i, q_i, S_{ij} , where all values need to be provided, even if 0, and the binning needs to be linear. This file can be obtained by Molecular Dynamics models or from experimental results. Using a complete mesh containing data from elastic, Bragg reflections, inelastic and deep inelastic scattering is quite difficult, for at present a model for a dynamic structure factor over several orders of energy and momentum transfers is not available. Moreover, while one can in principle use such file, its size would probably be too large and compromise the simulation efficiency.

The Impulse Approximation is, of course, an approximation. The use of a complex and adaptable component as the `Incoherent_SQW` can be machine-time inefficient when the $S(\mathbf{q}, \omega)$ is a mesh obtained by a simple analytic model as in our case (2.3). Even if one wants to use `Incoherent_SQW`, one needs to add an additional step to model the $S(\mathbf{q}, \omega)$ before running the McStas simulation. Every time we compute the $S(\mathbf{q}, \omega)$ for the `Incoherent_SQW` one needs to define *a priori* a range for energies and q-vectors, create a mesh and compute the result for each couple of values (\mathbf{q}, ω) . We present here a new component that can compute the $S(\mathbf{q}, \omega)$ directly inside McStas, taking as an input a configuration file, i.e., a text file containing easily retrievable information provided by the user about the atom properties. This new component is inspired by the DINSMS code by Mayers, Fielding and

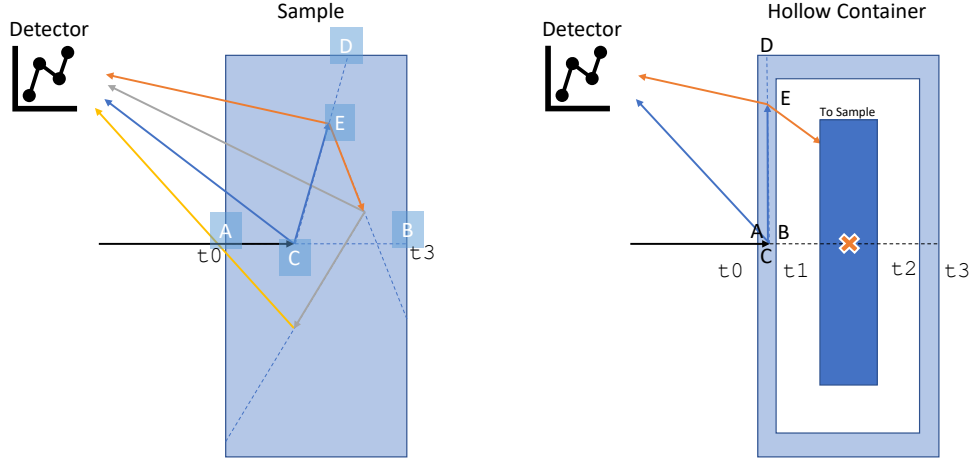


Figure 7.1: Schematic drawing of the geometry of interaction. The neutron-ray comes from the left. The first time the component is called the neutron propagates to the sample face (A). Distance AB is calculated and a point C of scattering is chosen. At this point the neutron could propagate to the detector or take a random direction: if this is the case the distance CD is calculated and a point of scattering E is selected. From this moment the choice is iterated with a Monte Carlo choice between escaping to the detector or interact again.

Senesi [41], separating the sample interaction from all the geometry and environment lines.

The correction for multiple scattering is an important step in the analysis of DINS experiments, and the use of the DINSMS code has proven necessary in a number of investigations over the past decades [42]. The main limitation of this latter code is that it can only model a homogeneous sample mixture, or approximate heterogeneous or structured specimen to a homogeneous mixture of atom species. If we want to include the multiple-scattering from an external container the procedure is to include the scattering centres from the cell inside the sample volume [43].

In this case, the McStas **components** are more useful with the integration of the *concentric* mode. Using this feature we can insert two distinct **components** inside our simulation: a cell with hollow geometry and the sample geometry. The main limitation is that all the sample vertexes must be inside the cell inner geometry.

We can summarize the interaction code in the following steps, refer to figure 7.1:

1. The neutron coming from the collimation region is moved to the sample face (point A);
2. The code computes the distance AB to the exit point in line with the neutron direction;

3. We choose a random point of scattering C depending on a probability function depending on the sample density and cross sections, this choice includes the secondary extinction;
4. The weight of the neutron-ray is multiplied by a factor mirroring the probability of scattering;
5. The code makes a decision according to a Monte Carlo procedure: scattering to the detector or multiple-scattering within the sample. The weight is corrected by the choice;
6. The weight is corrected so as to include the scattering probability according to the partial differential cross-section obtained within the impulse approximation;
7. If the neutron is scattered in the direction of the detector, then the weight is corrected by solid angle and transmission;
8. If the neutron is chosen to undergo multiple-scattering, than a new direction is computed and the steps 2-6 are repeated;

The McStas library includes intersection functions that return 1 or 0 if a neutron is between the boundaries of the defined geometry. This function return also two *time coordinates* (t_0, t_3) where the neutron intersects the geometry, considering the neutron in the origin of the axis (refer to figures 7.1 and 7.2).

The concentric mode is obtained using two calls to these functions. The second call uses as geometry the inner face of the hollow geometry. Suppose we are simulating a box-shaped sample. The first call of the intersection function takes as input the width, height and depth of the box. The function returns a `True` value if the neutron is inside this geometry. The second call of the intersection function takes as input the same dimensions but scaled down (the width for example is obtained by subtracting two times the thickness). This return values from this function are used to identify if the neutron is inside the hollow part.

In this component we have implemented all the common geometries, including cylinder; box; and sphere. One should note that the `off_intersect()` function can handle complex shapes but is not tested yet.

Listing 7.1: Using the mcstas library to calculate concentric mode

```

1  if (ia.shape == 0)
2      intersect = cylinder_intersect(&t0, &t3, x, y, z, vx, vy, vz, radius, yheight);
3  else if (ia.shape == 1)
4      intersect = box_intersect(&t0, &t3, x, y, z, vx, vy, vz, xwidth, yheight, zdepth);

```

```

5  else if (ia.shape == 2)
6      intersect = sphere_intersect(&t0, &t3, x, y, z, vx, vy, vz, radius);
7  else if (ia.shape == 3)
8      intersect = off_intersect(&t0, &t3, NULL, NULL, x, y, z, vx, vy, vz, offdata );
9  if (intersect) {
10     int flag_ishollow = 0;
11     if (thickness>0) {
12         if (ia.shape==0 && cylinder_intersect(&t1,&t2, x,y,z,vx,vy,vz, radius-thickness,yheight-2*
13             ↪ thickness))
14             flag_ishollow=1;
15         else if (ia.shape==2 && sphere_intersect (&t1,&t2, x,y,z,vx,vy,vz, radius-thickness))
16             flag_ishollow=1;
17         else if (ia.shape==1 && box_intersect(&t1,&t2, x,y,z,vx,vy,vz, xwidth-2*thickness, yheight-2*
18             ↪ thickness, zdepth-2*thickness))
19             flag_ishollow = 1;
20     }

```

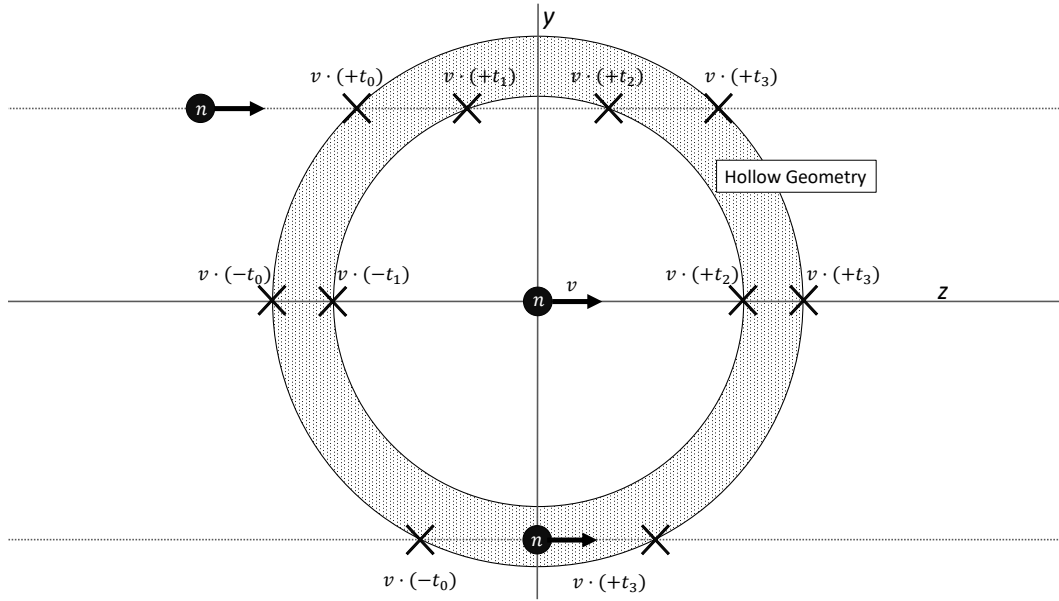


Figure 7.2: The interaction functions calculate the time coordinate considering the neutron in the origin of the axis. (top) The neutron is before the sample and all the times are positive, (center) in this case the neutron is inside the hollow part the time are showed with the explicit sign returned by the function, (bottom) the neutron is inside the sample and t_1 and t_2 are zero.

After running this code part we have four time coordinates (t_0, t_1, t_2 and t_3), using simple relations we can identify the neutron-ray position in the simulated sample. Referring to the right side of figure 7.1:

- If all times are positive we are before the sample, is the first interaction so we need to

propagate the neutron to the container face;

- If only t_0 is negative the neutron is inside the first part of the container. The neutron direction is *in going* to hollow part;
- If t_0 and t_1 are negative the neutron is in the hollow part so the code propagates it to the inner face of the container;
- If only t_3 is positive the neutron is inside the container and the direction is *outgoing*;
- If all times are negative the neutron passes completely the container;

The concentric mode basically duplicates the hollow component, replicating the related code before and after the inner component. Without this mode the same result can be obtained placing a cell `component` before and after the sample through the direction of the neutron. The cell `component` before the sample is easy to implement because of collimation, but the exit path of the neutron depends on scattering. Even applying the focus to the sample interaction, as we have done in our component, the solution is available only if the cell has very low scattering and interaction probabilities. If the cell contributes heavily to the neutron reaching the detectors we need to explicitly include all the components faces where the neutron goes through. If this is not possible the concentric mode is more accurate than using only two foils made from the same materials of the cell.

When the intersection variable is set to 1, we modify the velocity and the direction of each neutron ray according to the dynamic structure factor expressed in the impulse approximation. First of all we need to load into our components the physics variables describing the material. In most cases we are not interested on single species property, we want to study molecules or mixtures of different atoms so an input file format is required.

The adopted format this file is the following:

Listing 7.2: Format for input data, eg D2O

```
# AtomicMasses
2
# Total density (g/cm3)
1
# Mass (au), Xsect (b), absXsect (b), s.d. J(y), atoms in molecule
2.0135 7.64 0.000519 38 2
15.999 4.232 0.00019 38 1
```

We load the information in an `array structures` called `atom`. Passing it trough all the functions defined to simulate interaction. In appendix E are summarized the variable

name and abbreviation used in this work. First of all we need to initialize the variables. The bound-cross section from input file needs to be converted to free-scattering cross section via:

$$\mu(v) = n_{tot} \sum_{N_M} \left(\frac{n_a(M)4\pi b_M^2}{(1 + m/M)^2} + \frac{n_a \sigma_M^A}{v} \right) = n_{tot} \sum_{N_M} \frac{n_a 4\pi b_M^2}{(1 + m/M)^2} + n_{tot} \frac{\sum_{N_M} n_a \sigma_M^A}{v} \quad (7.1)$$

where M is the atom mass and $b_M = \sqrt{\sigma_s/4\pi}$, n_{tot} is the total density of molecules and n_a is the number of same atoms in the molecule, N_M is the number of different atom masses and v is the velocity of the incident neutron. In code form:

$$\mu(v) = \text{ia.my_s} + \frac{\text{ia.my_a_v}}{v} \quad (7.2)$$

the structure `ia` contains the aggregate parameters, `my_s` is initialized by the following:

```
1  ia.my_s = ia.num_dens*xstot(ia.NM, atom)*1E-22;
2  ia.my_a_v = ia.num_dens*xstot_a(ia.NM, atom)*1E-22;
```

where `ia.num_dens` is the number density of the molecule obtained by:

$$n_{tot} = \frac{\rho}{M} N_A \quad (7.3)$$

where ρ is the total density in g cm^{-3} , M is the molar mass of the molecule and N_A the Avogadro number. The factor 10^{-22} is needed to express the unit of measure in meters.

Listing 7.3: Function to calculate total cross section

```
1  double xstot(int NM, struct atomProp atom[])
2  {
3      double xstot = 0;
4      double G;
5      for (int i = 0; i < NM; i++)
6      {
7          G = 1.00867 / atom[i].mass;
8          xstot = xstot + 4.0 * PI * atom[i].na * pow(atom[i].b, 2) / (1 + G*G);
9      }
10     return xstot;
11 }
```

A similar function initializes `my_a_v`. The total scattering cross section maintains the v dependency and can be updated in every iteration.

Following the definition and initialization of the variables, the main code is contained in a `do{...}while(intersect)` loop. The `intersect` variable is set to 1 by the interaction function (listing 7.1) or by escape logic inside the run. After the check of neutron position

we initialize the weight multiplier (`p_mult`) to 1. While the simulation is running we update that value, applying to the real `p` value of the neutron-ray only when scatter happens. First we calculate the velocity of the neutron, the global variables of the neutron-ray are: `x,y,z,t,vx,vy,vz,sx,sy,sz,p` so we need to recover previous status reading this, and update them when writing a new status.

Listing 7.4: Setting the velocity, `ia` is a structure for not atom variables

```

1  p_mult = 1;
2  if (!v) v = sqrt(vx*vx + vy*vy + vz*vz);
3  if (v) my_a = ia.my_a*v*(2200/v);
4  my_t = my_a + ia.my_s; /* Macroscopic Xsect (tmp var) */

```

The `v` is initialized as 0 so we need to calculate it from the neutron parameters `v_x,v_y` and `v_z`. This calculation is performed only when the neutron reaches the sample faces from the collimation section. After every interaction in the loop of multiple-scattering the `v` is obtained by the final energy used as input for the double differential cross section. The `if (!v)` logic allows to enhance performance considering that a logic is generally quicker than a `sqrt` function.

In the total absorption cross section (`my_a`) we include the $2200/v$ dependency, considering the tabulated cross sections for thermal neutrons and velocities are expressed in m/s. At this point `ws` is ratio of neutron scattered from the neutron that has been scattered or absorbed and `d_path` is the distance from neutron position to the exit point from the sample (the segments AB or CD in figure 7.1).

```

1  /* Proba of scattering vs absorption (integrating along the whole trajectory) */
2  ws = ia.my_s/my_t; /* (inc+coh)/(inc+coh+abs) */
3  d_path = v * (dt0 + dt2); /* Length of full path through sample */

```

The time variables `dt0` and `dt2` are mutually exclusive. If the neutron is coming from an external component (*concentric mode*) `dt2` is set to 0. If the neutron is coming from the inner geometry `dt0` is set to 0. Considering `d_path` we store transmission and total scattering probabilities along this trajectory (`p_trans`, `p_scatt`). The `flag` is used alongside `mult_end` as escape command through the iteration, as discussed later

```

1  /* Proba of transmission/interaction along length d_path */
2  p_trans = exp(-my_t*d_path);
3  p_scatt = 1 - p_trans; /* portion of beam which scatters */
4  flag = 0; /* flag used for propagation to exit point before ending */
5  /* are we next to the exit ? probably no scattering (avoid rounding errors) */
6  if (ia.my_s*d_path <= 4e-7) {
7      flag = 1; /* No interaction before the exit */

```

If we maintain the physical value for scattering and transmission probabilities the code become very inefficient to compute results for materials with low scattering cross section. If the probability that a neutron is scattered is 1% , every 100 neutron simulated only 1 is going to contribute to the measure. We can force this probabilities using an opportune correction on the neutron weight. Suppose that we have 100 neutron-rays and each neutron ray is a collection of 100 neutrons, i.e. it has a weight of 100 neutrons. If the probability of scattering is 1% then only one neutron-ray reaches the detector that records 100 neutrons reached. The latter is the physics information, we can force more neutron-rays to interact but the neutrons that the detector counts need to remain 100. Consider w the neutron-ray weight and P the probability associated to a phenomenon during a Monte Carlo simulation. If we use P_{MC} as the Monte Carlo probability for the neutron-ray interaction than the following rules is applied

$$P_{MC}w = P \quad (7.4)$$

if we want to force the 2% of the neutron-rays to interact

$$(0.02)w = (0.01) \rightarrow w = 0.5 \quad (7.5)$$

So two neutron-rays reach the detector carrying 50 neutrons each. Following this law we can set a value from 0 to 1 from McStas component in the variable `p.interact` forcing the scattering probabilities.

```

1  /* force a given fraction of the beam to scatter */
2  if (p.interact>0 && p.interact<=1) {
3      /* we force a portion of the beam to interact */
4      /* This is used to improve statistics on single scattering (and multiple) */
5      if (!SCATTERED) mc.trans = 1-p.interact;
6      else          mc.trans = 1-p.interact/(4*SCATTERED+1); /* reduce effect on multi scatt */
7  } else {
8      mc.trans = p.trans; /* 1 - p.scatt */
9  }
10 mc.scatt = 1 - mc.trans; /* portion of beam to scatter (or force to) */
11 if (mc.scatt <= 0 || mc.scatt>1) flag=1;

```

In the following code `mc.scatt` and `mc.trans` are the new values for scattering and transmission probabilities. If we are at the end of the multiple-scattering iterations the `mult_end` is 1 so the `flag` is set to 1. If this is true the neutron escapes to the detector and the weight is corrected by the transmission probabilities and Monte Carlo choice. If we are in concentric mode the `target_index` is undefined so the neutron propagates along the z axis.

The propagation is assigned to the macro `PROP_DT(x)` that modifies the neutron parameters according to its velocity and position, using the time `x`. If the `flag` is 0, the interaction choice is made. If the interaction happen we update the weight trough the `p_mult` variable, retain the scattered fraction from `ws` and correct by the Monte Carlo choice through `p_scatt/mc_scatt`.

```

1  /* MC choice: Interaction or transmission ? */
2  if (mult_end) flag = 1;
3  if (!flag && mc_scatt > 0 && (mc_scatt >= 1 || rand01() < mc_scatt)) { /* Interaction neutron/
   ↪ sample */
4      p_mult *= ws; /* Update weight ; account for absorption and retain scattered fraction */
5      /* we have chosen portion mc_scatt of beam instead of p_scatt, so we compensate */
6      if (!mc_scatt) ABSORB;
7      p_mult *= fabs(p_scatt/mc_scatt); /* lower than 1 */
8  } else {
9      flag = 1; /* Transmission : no interaction neutron/sample */
10     if (!mc_trans) ABSORB;
11     /* attenuate beam by portion which is scattered (and left along) */
12     /* If is the end of multiple-scattering than do not correct MC */
13     if(mult_end){
14         p_mult *= fabs(p_trans);
15     }else{
16         p_mult *= fabs(p_trans/mc_trans);
17     }
18 }
19 if (flag | mult_end) { /* propagate to exit of sample and finish */
20     intersect = 0;
21     // if (p_mult<0) ABSORB;
22     p *= p_mult; /* apply absorption correction */
23     PROP_DT(dt0+dt2);
24     break; /* exit main multi scatt while loop */
25 }

```

Using the total scattering cross sections (absorption and scattering) obtained from the listing 7.3 we calculate the interaction point inside the sample.

The probability that the neutron will scatter between x and $x + dx$ is $\mu \exp(-\mu x)dx$. Calling d the total possible path length of the neutron in the sample, the probability that the neutron will scatter within the distance d is:

$$W_d = \int_0^d \mu \exp(-\mu x)dx = 1 - \exp(-\mu d) \quad (7.6)$$

The probability that it will scatter somewhere between 0 and x , given that it scatters

somewhere in the distance d is:

$$C(x) = \frac{1 - \exp(-\mu x)}{1 - \exp(-\mu d)} \quad (7.7)$$

So considering $C(x)$ a randomly generated value z from 0 to 1 we can invert and obtain:

$$x = -\frac{1}{\mu} \log(1 - z(1 - e^{-\mu d})) \quad (7.8)$$

In the code we generate directly a value between 0 and B using `rand0max(B)`. Writing:

$$dt = -\frac{1}{my_t} \log[1 - \text{rand0max}(1 - e^{-my_t \cdot d_path})] \quad (7.9)$$

Listing 7.5: Calculate the interaction path

```

1  /* Select a point at which to scatter the neutron, taking
2     secondary extinction into account. */
3  if (my_t*d_path < 1e-6)
4  /* For very weak scattering, use simple uniform sampling of scattering
5     point to avoid rounding errors. */
6     dt = rand0max(d_path); /* length */
7  else
8     dt = -log(1 - rand0max((1 - exp(-my_t*d_path)))) / my_t; /* length */
9  Li = dt; /* Penetration in sample: scattering+abs */
10 dt /= v; /* Time from present position to scattering point */
11 /* If t0 is in hole, propagate to next part of the hollow cylinder */
12 if (dt1 > 0 && dt0 > 0 && dt > dt0) dt += dt1;
13 PROP_DT(dt); /* Point of scattering*/
14 /* Escape variabel for multiple-scattering */
15 mult_end=0;

```

We are basically choosing the point C in fig 7.1. Using the previous formula we can update the weight using the integrated scattering probability `p_scatt` along `d_path`. At this point we propagate the neutron to the point of scattering and initialize the `mult_end` variable. At this moment the real interaction takes place and the multiple-scattering feature is defined. Through Monte Carlo we select two types of interactions, if the neutron is going to the detector the output energy and velocity versor are set considering the detector direction and efficient energy range (the neutron pass a resonant foil of Gold so we do not need neutrons with energy far away from resonance energy). If we are going to iterate the interaction final values are randomly generated assuming isotropic scattering.

Listing 7.6: Scattering to detector

```

1  /* Choice: detector or multiple-scattering?*/

```

```

2  if(rand01() < 0.5){
3      mult_end=1;
4      if ((ia.tx || ia.ty || ia.tz)) {
5          aim_x = ia.tx-x;
6          aim_y = ia.ty-y;
7          aim_z = ia.tz-z;
8      }
9      if(ia.aw && ia.ah) {
10         randvec_target_rect_angular(&vx_f, &vy_f, &vz_f, &solid_angle,
11             aim_x, aim_y, aim_z, ia.aw, ia.ah, ROT_A_CURRENT_COMP);
12     } else if(ia.xw && ia.yh) {
13         randvec_target_rect(&vx_f, &vy_f, &vz_f, &solid_angle,
14             aim_x, aim_y, aim_z, ia.xw, ia.yh, ROT_A_CURRENT_COMP);
15     } else {
16         randvec_target_circle(&vx_f, &vy_f, &vz_f, &solid_angle, aim_x, aim_y, aim_z, focus_r);
17     }
18     NORM(vx_f,vy_f,vz_f);
19     th0 = acos((vx_f*vx_f+vy_f*vy_f+vz_f*vz_f)/v);
20     if (E1Active==0){
21         e1range(th0,E_i,ia.NM,atom,&E1MIN,&E1MAX);
22         E_f=E1MIN+rand01()*(E1MAX-E1MIN);
23     }else{
24         E_f=finalEnergyAuDD(rand01());
25     }
26     if(E_f<0){
27         printf("E_f is less than zero. Absorb.\n");
28         ABSORB;
29     }
30     pdcs_weight = pdcs(E_i,E_f,th0,ia.NM,atom,ia.FSE);
31
32     if(E1Active==0){
33         p_mult *= pdcs_weight*solid_angle*(E1MAX-E1MIN)/ia.xst;
34     }else{
35         p_mult *= pdcs_weight/ia.xst;
36     }
37     v=SE2V*sqrt(E_f);
38     vx = v * vx_f;
39     vy = v * vy_f;
40     vz = v * vz_f;
41     /* Correction by mc choice */
42     p *= p_mult/0.5;
43 }else{
44     ...
45 }

```

The first 18 lines are defining the velocity versor pointing to the detector. From McStas we can retrieve the position of the detectors (**tx**,**ty**,**tz**) stored in the **ia** structure.

The function called **randvec_target_rect_angular** etc. generates final velocity vector components **vx_f**,**vy_f**,**vz_f** pointing to a random position inside a rectangular (**aw**,**ah**) or circular imposed area. This function returns also the **solid_angle** covered by this generation. We don't need the exact final velocity provided by this functions, we need only the aiming capabilities, so we normalize through the macro **NORM**. At this point we calculate the angle of scattering (**th0**) and call the function **eifinal** explained later in this chapter. This function generates the final energy range through (**E1MIN**,**E1MAX**) from which we choose randomly a value **E_f**. This is a uniformly distributed choice inside the range. If we are simulating only back scattering detectors than we can generate E_f taking into account the gold foil resonance. A detailed description of the difference between the energy selection mechanisms is reported in Appendix B. We are generating x with uniform probability between a and b , so we need to weight the event by the probability that the event would occur. Using the following to generate x :

$$x = a + (b - a)z \quad (7.10)$$

where $0 < z < 1$ is a random number. The weight is set as

$$w(x) = (b - a)P(x) \quad (7.11)$$

The factor $(b - a)$ ensures that the average weight given to the event is 1. We can see it from the integral mean value theorem:

$$\int_a^b P(x)dx = \int_a^b \bar{P}dx = (b - a)\bar{P} = 1 \quad (7.12)$$

Following these, the neutron-ray weight must be set:

$$w = (E_{1max} - E_{1min}) \frac{d^2\sigma(E_0, E_1, \theta)}{d\Omega dE_1} \bigg/ \frac{d\sigma(E_0, \theta)}{d\Omega} \quad (7.13)$$

where

$$\frac{d\sigma(E_0, \theta)}{d\Omega} = \int \frac{d^2\sigma(E_0, E_1, \theta)}{d\Omega dE_1} dE_1 \quad (7.14)$$

$$\left(\frac{d\sigma(E_0, \theta)}{d\Omega} \right) w = (E_{1max} - E_{1min}) \frac{d^2\sigma(E_0, E_1, \theta)}{d\Omega dE_1} \quad (7.15)$$

integrating on solid angle, considering an isotropic scattering with uniform probabilities and using again the integral mean value theorem on θ :

$$\sigma(E_0)w = 4\pi(E_{1max} - E_{1min}) \frac{d^2\sigma(E_0, E_1, \theta)}{d\Omega dE_1} \quad (7.16)$$

So the weight must be corrected by:

$$w = \frac{1}{\sigma(E_0)} 4\pi(E_{1max} - E_{1min}) \frac{d^2\sigma(E_0, E_1, \theta)}{d\Omega dE_1} \quad (7.17)$$

If *concentric* is active **tz** is set to 1. Using the final energy we call the function **pdcs** that calls the impulse approximation and returns the double differential cross section and stores it in **pdcs_weight**. The weight of the event is updated using 7.13 and the new velocity is stored and applied to the neutron variables. Correcting again for the probability of emission inside the solid angle of the detector and the montecarlo choice we apply to the neutron weight **p** the correction calculated in **p_mult**. When this choice is made the simulation is going to end, **mult_end** is set to 1 so the flag triggers. The second iteration calculates the new **p_trans** values from the scattering position and the neutron propagates outside the sample. If the choice is *multiple-scattering* than the code is similar but without focal vector. We chose a random direction of scattering with the polar angle θ and the azimuthal angle ϕ and generate a new unitary vector.

Listing 7.7: Scattering for multiple-scattering

```

1      ...
2      }else{
3      th= acos(2.0*rand01()-1.0);
4      ph= 2.0*PI*rand01();
5
6      vx_f = cos(th)*cos(ph);
7      vy_f = cos(th)*sin(ph);
8      vz_f = sin(th);
9
10     th0 = acos((vx_f*vx+vy_f*vy+vz_f*vz)/v);
11
12     E_i = VS2E*v*v;
13     eIrange(th0,E_i,ia.NM,atom,&E1MIN,&E1MAX);
14     E_f=E1MIN+rand01()*(E1MAX-E1MIN);
15
16     if(E_f<0){
17         printf("Multi Final Energy Negative! ABSORB\n");
18         ABSORB;
19     }
20
21     pdcs_weight=pdcs(E_i,E_f,th0,ia.NM,atom,ia.FSE);
22     // E_f+=rand.boltz;
23
24     p_mult *= pdcs_weight*4.0*PI*(E1MAX-E1MIN)/ia.xst;

```

```

25
26     v=SE2V*sqrt(E_f);
27
28     vx = v * vx_f;
29     vy = v * vy_f;
30     vz = v * vz_f;
31
32     p *= p_mult/0.5;
33 }

```

7.1 Special Functions

The `pdcs` and `e1final/e1range` functions are where the interaction and efficiency resides. The energy range is selected dynamically during the run-time of the code considering the final energy sustainable by impulse approximation. Depending on masses and variance of $J(y)$ of the atoms inside the sample component we select the final energy to not modify to zero the weight of the neutrons. The listing is showed in 7.8. From the velocity of the neutron we calculate E_0 , the code then converts it to wave vector k_0 . We initialize a large energy range and than we update it considering the most valuable range for impulse approximation. From theory we know that initial and final velocities are correlated by kinematics if the energy transfer is exactly the recoil energy. If θ is the angle between k_0 and k_1 we use the relation $v_1 = \alpha v_0$ with

$$\alpha = \frac{\cos \theta + \sqrt{(M^2) - \sin^2 \theta}}{M + 1} \quad (7.18)$$

Knowing v_0 we can evaluate v_1 and so E_1 and then the recoil energy:

$$\hbar\omega = E_0 - E_1 \quad (7.19)$$

from k_0 and k_1 we can calculate $|\mathbf{q}|$:

$$q = \sqrt{k_0^2 + k_1^2 - 2k_0k_1 \cos(\theta)} \quad (7.20)$$

From $J(y)$ variance we can calculate the final energy variance:

$$\sigma_{energy} = \frac{\hbar q}{2M} \sigma_J; \quad (7.21)$$

so we select a final energy range for our final energy on $10\sigma_E$ from recoil energy. For `e1final` the code is similar, but instead of $E_0 - \hbar\omega$ we use the final energy around the gold absorption resonance, 4907eV with a range of $\pm 5\sigma_E$.

Listing 7.8: Function to choose final energies

```
1 int elrange(double th, double E0, int NM, struct atomProp *atom, double *E1MIN, double *E1MAX)
2 {
3     double k_0, k_1;
4     double E1, E1A, E1B;
5     double qrr, qr, wr, wid;
6     double alpha;
7
8     /* From neutron incident energy calculate k_0 wave vector*/
9     k_0 = sqrt(E0 / 2.0717);
10    /* Set large interval for final energy*/
11    *E1MIN = 1E10;
12    *E1MAX = -1E10;
13    /* From the recoils energy of the sample atoms choose the new range of energy*/
14    for (int i = 0; i < NM; i++)
15    {
16        /* Alpha is used to obtain k_1 knowing theta (th) and atom mass, only in Impulse Approx*/
17        alpha = (cos(th) + sqrt(pow(atom[i].mass, 2) - sin(th) * sin(th))) / (atom[i].mass + 1);
18        k_1 = alpha * k_0;
19        /*hbar/2 = 2.0717 using meV*/
20        E1 = 2.0717 * k_1 * k_1;
21        /* Q square */
22        qrr = k_0 * k_0 + k_1 * k_1 - 2.0 * k_0 * k_1 * cos(th);
23        if (qrr < 0.0) qrr = 0;
24        qr = sqrt(qrr);
25        /* Recoil, using E1 from alpha. Choosing energies in 10 sigma from recoil, scaled */
26        wr = E0 - E1;
27        wid = 2.0717 * atom[i].st_dev * qr / atom[i].mass;
28        E1A = E0 - wr - 10.0 * wid;
29        E1B = E0 - wr + 10.0 * wid;
30        /* Setting */
31        if (E1A < *E1MIN) *E1MIN = E1A;
32        if (E1B > *E1MAX) *E1MAX = E1B;
33    }
34    if (*E1MIN < 0.0) *E1MIN = 0.0;
35    return 0;
36 }
```

For the partial differential cross-section we recover the formalism from 3.15

$$\frac{d^2\sigma_M}{d\Omega dE_1} = |b_M|^2 \sqrt{\frac{E_1}{E_0}} S_M(\mathbf{q}, \omega) \quad (7.22)$$

again we use the West scaling, using the standard deviation of $J(y)$ to write the $S(\mathbf{q}, \omega)$ as

$$S(q, \omega) = \frac{M}{\hbar^2 q} J(y) \quad (7.23)$$

considering $\hbar\omega = E_1 - E_0$ we write

$$\begin{aligned} y &= \frac{M\hbar\omega}{\hbar^2 q} - \frac{q}{2} \\ &= \frac{M\hbar\omega}{\hbar^2 q} - \frac{M}{\hbar^2 q} \frac{\hbar^2 q^2}{2M} \\ &= \frac{M}{\hbar^2 q} \left(\hbar\omega - \frac{\hbar^2 q^2}{2M} \right) \end{aligned} \quad (7.24)$$

to include the final state effects a variable (FSE) is passed by the instrument code. At the moment the correction includes an asymmetric contribute only from the third order Hermite Polynomials:

$$H_3 \left(\frac{y}{\sigma_y \sqrt{2}} \right) = 8 \left(\frac{y}{\sigma_y \sqrt{2}} \right)^3 - 12 \left(\frac{y}{\sigma_y \sqrt{2}} \right) \quad (7.25)$$

from which we obtain

$$J(y)_{fse} = J(y) - A_3(y)J(y) \quad (7.26)$$

where

$$A_3(y) = \frac{\sigma_y}{3q\sqrt{2}^3} H_3 \quad (7.27)$$

Listing 7.9: Function to write PDCS

```

1 double pdcS(double E_0, double E_1, double th, int NM, struct atomProp atom[], int FSE)
2 {
3     double RT2PI;           // sduareroot of PI
4     double k_0, k_1, q;     // wave vectors
5     double w ;              // E_i - E_f energy
6     double pdcS, sqw;       // Partial differential cross section, S(q,w)
7     double Y_west, J, h3, A3; // J(y) and Hermite correction
8
9     RT2PI = sqrt(2.0 * acos(-1.0));
10    /* From Energies in input calculate Q and W*/
11    k_0 = sqrt(E_0 / 2.0717);
12    k_1 = sqrt(E_1 / 2.0717);
13    q = sqrt(k_0 * k_0 + k_1 * k_1 - 2.0 * k_0 * k_1 * cos(th));
14    w = E_0 - E_1;
15    pdcS = 0.0; //initialize pdcS
16    /* For each atom in input calculate contributes to PDCS */
17    for (int i = 0; i < NM; i++)
18    {

```



```

19  /* Avoiding q=0 errors */
20  if (q != 0)
21  {
22      /* From mass and W generate the Y variables from West-Scaling */
23      Y_west = atom[i].mass * w / (4.18036 * q) - q / 2.0;
24      /* Leading term (gaussian) for J(Y) */
25      J = exp(-(Y_west * Y_west) / (2.0 * atom[i].st_dev * atom[i].st_dev)) / (atom[i].st_dev * RT2PI);
26      /* If final state effects are active, correct the J(Y) with Hermite Polynomials */
27      if (FSE==1){
28          h3=8*pow(Y_west/(atom[i].st_dev*sqrt(2)),3)-12*Y_west/(atom[i].st_dev*sqrt(2));
29          A3=atom[i].st_dev/(3*q*pow(sqrt(2),3))*h3;
30          J*=(1-A3);
31      }
32      /* Calculate Partial Differential Cross Section */
33      sqw = atom[i].mass * J / (4.18036 * q);
34      pdcs = pdcs + pow(atom[i].b, 2) * (k_1 / k_0) * sqw;
35  }
36  else
37  {
38      pdcs = pdcs + pow(atom[i].b, 2);
39  }
40  }
41  return pdcs;
42  }

```

7.2 BackScattering Results for D2O

In order to highlight the results from our simulation, we tried to replicate a complex measure performed on the VESUVIO spectrometer [43]. The work needed a really precise measure of the momentum distribution of deuterium. In presence of such light atoms, the multiple-scattering contribution is widely distributed across the time of flight spectrum, and the oxygen and copper contributions are added to this background. The simulation of the correction for such structured multiple-scattering, even with the DINSMS code, is quite tricky. There is the need to create a fictitious homogeneous sample inside the software that contains the same amount of atoms as the ones involved during the experiment. Even counting how many atoms of copper we need to include in our sample, in order to reproduce the cell-container contribution to the signal, is not straightforward. Simply adding the atoms does not work, there is the need to fine tune the parameters in order to estimate the correct intensity of the signal. Not all the atoms in the cell-container contribute at the same way. Some atoms of the copper are directly illuminated by the neutron flux, while some others will interact only with multiple-scattered neutrons. The McStas simulation can be useful in this case for it provides insightful graphical information, even if these preliminary results show large discrepancies. In our simulation we used two impulse-approximation components, one for the sample and one for the sample container. The sample was $7 \times 7 \times 0.5$ cm³, the copper container was set to match this dimension considering a thickness of 0.5 mm.

The input parameters are showed in table 7.1. In figure 7.3 a neutron interaction is showed. We can see that there are more neutrons coming at the same time at sample position. This is done by using the **SPLIT** command where a neutron-ray that has reached a component is replicated as many times as defined in the split command (10 by default). Since our component calls some random variables in the Monte Carlo process (like direction, scattering probabilities and so on) every copy of the neutron-ray interacts differently enhancing the resolution of the signal. A comparison of the simulation and experimental results is showed in figure 7.4.

In figure 7.4 we can see the comparison between the simulation and the experimental result for the detector S44. The multiple-scattering contribution is heavily underestimated and the peaks positions are shifted to lower time-of-flight values. The copper contribution seems to be over-estimated, but this can be justified by the sequential nature of McStas simulations. A good number of neutron-rays reach the second face after the transmission within the sample. This neutron-rays interact with the copper face and are scattered in the direction of the detector. Since this component is positioned after the sample component

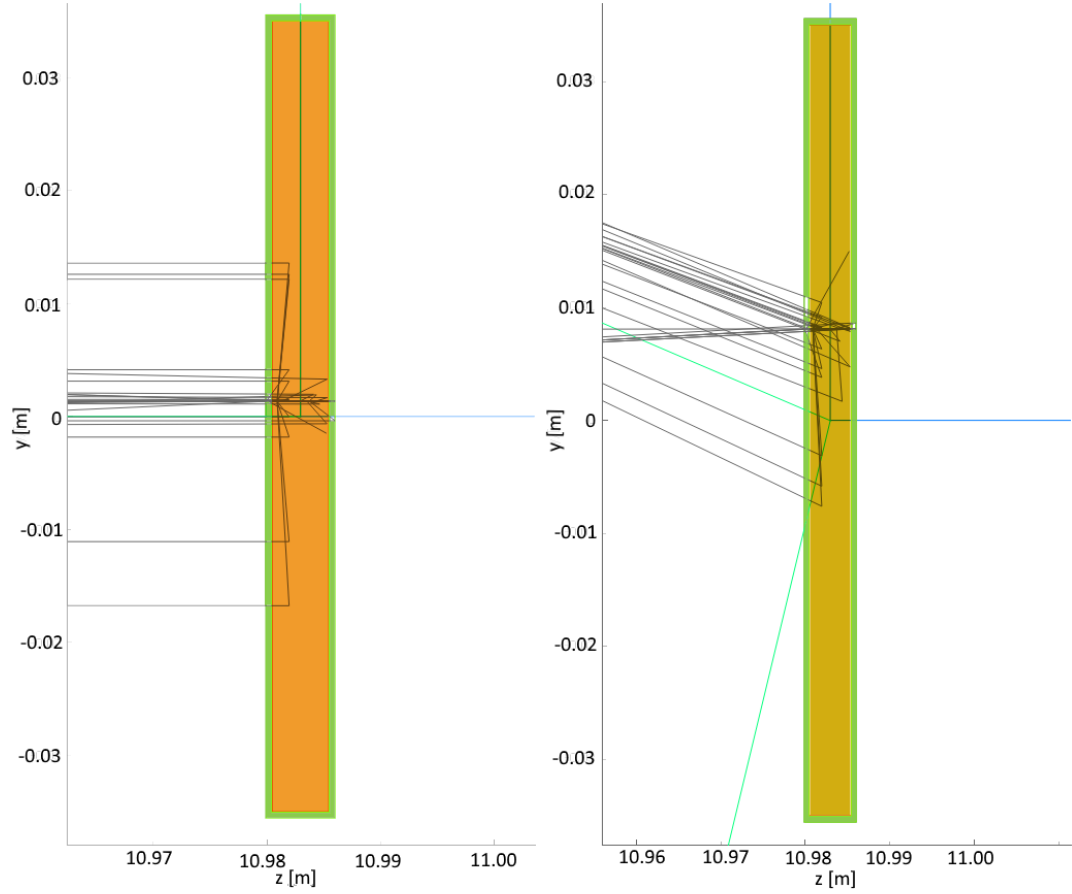


Figure 7.3: Graphical visualization of the sample component in the McStas simulation. (left) Side view, (right) Top view. Neutrons are propagated along z . The green figure is the sample container, and the orange one is the sample itself. This simulation makes use of the `SPLIT` command where the neutrons interacting within the geometry are replicated. Every path is shown as the black lines.

<pre> *****Impulse Approximation: CuIn ***** * Atomic Species Number: 1 * Total density: 8.960 [g/cm3] * * # Mass (au), Xsect (b), absXsect (b), s.d. J(y), atoms in ↳ molecule * 63.55 8.030 0.550 20.000 1.000000 * * Total Molar Mass: 63.546 * Total Num Molecular density: 8.49103e+022 [1/cm3] * Concentric: Total Number of Molecule in V: 5.91468e+023 * * Mass: 63.546 -> Atom number density: 8.49103e+022 [1/ ↳ cm3] * Concentric: Total Number of Atoms in volume: 5.91468e ↳ +023 * Atom Conc: 1.00 * * Total Penetration Depth Length: 1.41 [cm] * Total free scattering xsect= 7.781 [b] * * Final state effects: ACTIVE * * Multiple Scattering Max Order: 2 * multiple-scattering from order: 0 * ***** </pre>	<pre> *****Impulse Approximation: D2O ***** * Atomic Species Number: 2 * Total density: 1.110 [g/cm3] * * # Mass (au), Xsect (b), absXsect (b), s.d. J(y), atoms in ↳ molecule * 2.01 7.640 0.001 7.000 2.000000 * 16.00 4.232 0.000 12.000 1.000000 * Total Molar Mass: 20.026 * Total Num Molecular density: 3.33787e+022 [1/cm3] * Total Number of Molecule in V: 8.17778e+023 * * Mass: 2.014 -> Atom number density: 6.67574e+022 [1/ ↳ cm3] * Total Number of Atoms in volume: 1.63556e+024 * Atom Conc: 0.67 * * Mass: 15.999 -> Atom number density: 3.33787e+022 [1/ ↳ cm3] * Total Number of Atoms in volume: 8.17778e+023 * Atom Conc: 0.33 * * Total Penetration Depth Length: 2.85 [cm] * Total free scattering xsect= 10.527 [b] * Final state effects: ACTIVE * Multiple Scattering Max Order: 2 * multiple-scattering from order: 0 * ***** </pre>
--	--

Table 7.1: Input parameters for the D2O simulation. The copper parameters on the left side are replicated for the outgoing neutrons for the concentric mode.

inside the code (the second face correspond to the outgoing neutron in concentric mode) there are no additional interaction with the sample on the path through the detector.

Comparing only the multiple-scattering contribution (figure 7.5) from McStas with the DINSMS simulations we can see that the deuterium and oxygen peaks (first and second ones from left) show comparable intensities. This is clearly wrong due to experimental measure and the DINSMS simulation. Probably one needs to refine the partial differential cross section to better consider molecular density.

In figure 7.6 we tried to show the differences between a homogeneous sample simulation and one with distinct geometry for sample and container. The homogeneous sample configuration was obtained by maintaining the same number of atoms, calculated from the distinct geometries configuration (see input files 7.1) . Even if the multiple-scattering is not well handled we can see that between the copper and oxygen peaks the mixed configuration seems to raise the counts. This can be explained by the fact that there is more inter-atomic scattering between copper and sample components. In the distinct geometry this is a more rare event since only the sample atoms near the borders tend to multiscatter with the copper.

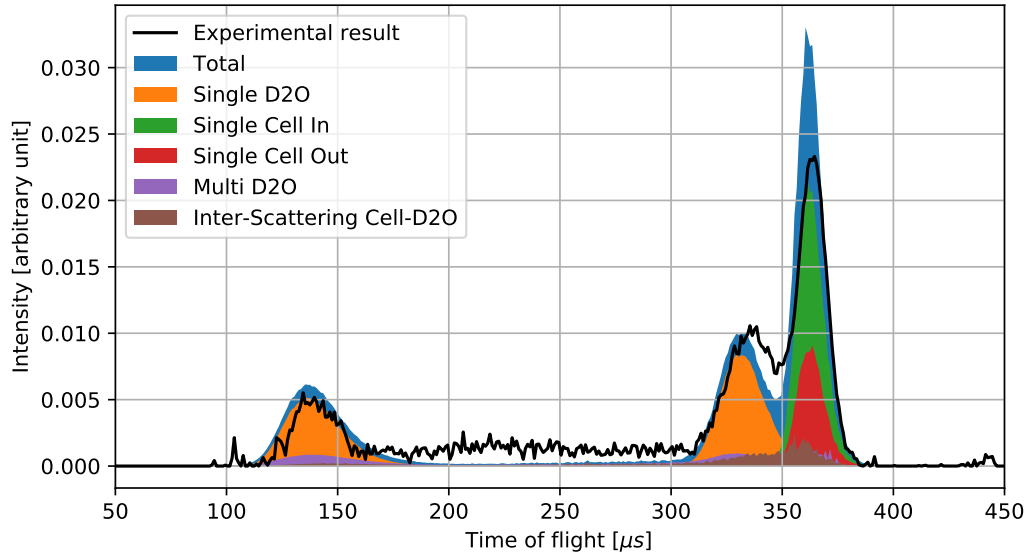


Figure 7.4: Simulation and experimental results for the detector S44. The data is normalized by area. The multiple-scattering contribution between 150 and 300 μs is not well reproduced. Some useful implementations are showed: McStas can split the signal contribution from cell and sample, we are even capable of showing the single faces of cell contribution (red and green). The brown area correspond to the neutrons that have scattered in the cell and in the sample.

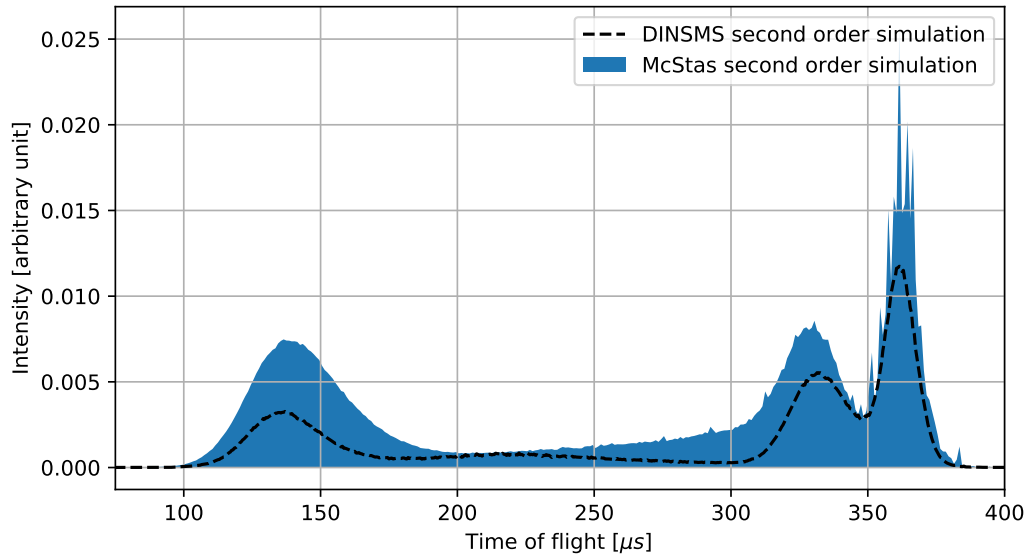


Figure 7.5: The only multiple-scattering contribution is showed. The intensities needs to be corrected. The results are normalized by total area. The McStas simulation show a more multiple-scattering contribute from the oxygen (third peak from left) than the DINSMS code.

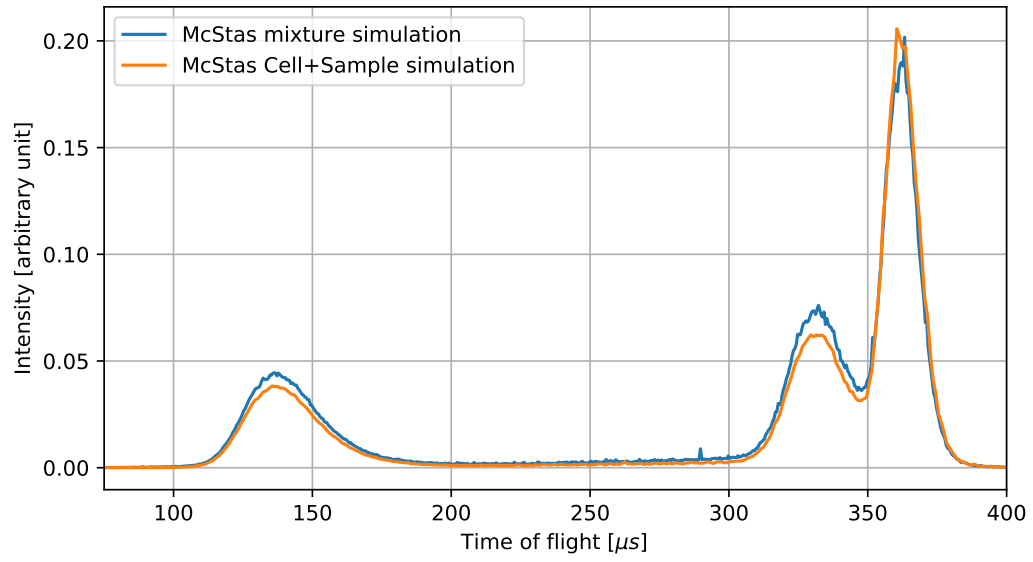


Figure 7.6: Total scattering signal from McStas code. The orange is the simulation as showed before. The blue line is the same simulation but with an homogeneous sample configuration.

Chapter 8

Conclusion

We have successfully implemented the VESUVIO spectrometer into McStas software package. It is necessary to work on some fine tuning that will be carried out opening a collaboration with the McStas developers and instrument scientists.

The collimation section is well reproduced. The distribution of the neutron energy spectrum coming from the moderator agreed very well with the experimental measure [22]. The intensity discrepancies are comparable with the same values obtained by other instruments on Target Station 1 (Ref [38]). Since the main upgrade to this section relies on a better simulation model for the Target Station these results will be sent to the ISIS Neutronic Group, the developers of the source component `Commodus_I`, to enhance the MCNPX simulation of the target station. The already existing components of McStas were tested with a larger energy ranges, this exposed some limitations for more complex `components` (like the sample ones). In order to avoid this limitation a new `component` was developed. Single scattering results are in agreement with the existing simulation tool (DINSMS), but multiple-scattering contributions seem to be underestimated.

Heteronuclear samples are well described and the component can handle concentric geometries. There is an overestimation of cell contribution, this is due to the sequentially nature of the McStas software. A neutron that reaches the backside of the cell is scattered along the detector direction without interacting again within the sample. This problem can be avoided in future rewriting this new component as an `UNION` component. The `UNION` components are a special set of `components` written by Bertelsen [44]. In this case the multiple-scattering contribution, transmission, and absorption processes are carried out by the code. The developer writes down only the physics of a single scattering process, without iterating or optimizing the neutron-rays flux, and assigns this physics to a geometry

component. The code itself then computes all the neutron multiple scattering between the geometries and optimizes the sequential calculation. For this reason we think that a complete description inside McStas can bring a solid help to the instrument scientists for the VESUVIO multiple-scattering analysis. Another limitation of DINSMS can be bypassed easily: The capability of the McStas model to use **filter_gen** components for resonant foils permits to refine the front scattering resolution. We can selectively analyze each foil contribution to the signal and easily change its geometry or thickness to minimize resolution routines. Once the new component is refined every upgrade of the VESUVIO spectrometer can be simulated and highly optimized without rewriting the entire simulation. After the future validations, the component will be sent to McStas developers in order to be included in future releases.

Appendix A

SQW Algoritm

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os

###from user
M=207.2 #Lead Atomic mass (amu)
NM=1.67*10**(-27) #Neutron Mass
SIG=38 #standard deviation of J(y)
os.chdir("path/to/directory")

RT2PI=np.sqrt(2.0*np.arccos(-1.0))
Q=np.arange(1,600,1)
W=np.arange(1,1000,1)
WW,QQ=np.meshgrid(W,Q)
Y=M*WW/(4.18036*QQ)-QQ/2.0
J=np.exp(-Y**2/(2*np.power(SIG,2)))/(SIG*RT2PI)
SQW=M*J/(4.018036*QQ)

# %% Writing to MCSTAS format
file=open('Pb2.sqw','wb')
intro=(['# Format: Sqw data file for Isotropic_Sqw <http://www.mcstas.org>',
        '# Simulated using impulse approximation',
        '# Using SQW_generator from A. Di Giulio',
        '# title: Pb: S(q,w) impulse approximation',
        '# Date: 22/10/2017',
        '# filename: Pb.sqw',
        '# format: Sqw data file for Isotropic_Sqw (McStas)',
        '# signal: Min=-7.84046e-08; Max=1.73437; Mean=0.00197862; sum=3489.95;'],
```

```

'## type: array_2d'+str(SQW.shape),
'## xylimits: '+str(Q.min())+' '+str(Q.max())+' '+str(W.min())+' '+str(W.max()),
'## xlabel: Wavevector [Angs-1]',
'## ylabel: Energy [meV]',
'##',
'## Physical parameters:',
'## Comment Rewritten in Python from FORTRAN code using standard deviation of J(y)='+str
    ↳ (SIG),
'## Instrument VESUVIO at ISIS',
'## Final energy 4.9 [eV] Neutron final energy used for the measurement',
'## Material Pb' ,
'## Phase none',
'## Scattering total process',
'## T_b 2022 [K] Boiling T',
'## T_m 600.61 [K] Melting T',
'## Temperature 300 [K]',
'## classical 1 [0=from measurement, with Bose factor included, 1=from MD, symmetric]',
'## dT 20 [K] T accuracy',
'## density 11.3 [g/cm3] Material density',
'## sigma_abs 0.171 [barn] Absorption neutron cross section',
'## sigma_coh 11.115 [barn] Coherent scattering neutron cross section',
'## sigma_inc 0.003 [barn] Incoherent scattering neutron cross section',
'## weight 207.2 [g/mol] Material weight',
'##'])
np.savetxt(file, intro,fmt='%s')
np.savetxt(file, Q[np.newaxis],fmt='%.7g', delimiter=' ',header='Q values in AA-1')
np.savetxt(file, W[np.newaxis],fmt='%.7g', delimiter=' ',header='E values in meV',newline="\n
    ↳ ")
np.savetxt(file,SQW,fmt='%.5g', delimiter=' ',header='SQW matrix',newline="\n")
file.close()

```

Appendix B

Gold Foil

While most thermal neutron spectrometer can be easily converted into McStas without intensive effort, the VESUVIO spectrometer needs a more detailed description and a bit more complicated coding machinery. In an inverse geometry instrument the final energy is fixed. In the VESUVIO spectrometer is based around resonant filters, in the back-scattering case they are used as absorbing foils and in forward-scattering as γ -emitting foils. The absorption is already implemented into McStas via the `filter_gen` component that loads a two dimensional ASCII table with the weight correction factor. For the forward-scattering foils we need a more deep implementation because McStas can not handle at this moment the photons radiation.

As already seen the peculiarity of VESUVIO is the changing of the foil configuration to filter more or less neutrons or emit more or less γ radiation. There are a total of 6 configuration setup, and all of that contribute to obtain the final Compton Profile from the sample.

B.1 Absorption foils

The foils in back-scattering are mounted on a disk divided in three dials, one without foils, one with a thin foil and one with a thick foil. In one run of VESUVIO every two periods of measure the disk rotates so every detector writes down three different measure corresponding to the foil faced. The implementation of this movement inside the code depends strongly on the simulation focus. Using the `filter-gen` component we enhance the modular nature of McStas and simplify the analysis of resolution by changing the resonant material, or the β parameter optimization. This method can be performed by two setups:

- Insert three `filter_gen` components with no foil, thick foil and thin foil tables;
- Insert only one `filter_gen` component using a table with the double difference corresponding table.

If we do not need to change frequently the foils material the most efficient way is to write down directly inside our sample component the code assigned to choice the final energy, considering the probability of emission taking and into account the double difference result.

We are going to show results for all three cases. Showing how resolution and performance change for each configuration.

B.1.1 Three `filter_gen`

In these simulations we are going to insert between sample and detector two `Filter_gen` component that load two custom made `.dat` files, created using a python script, Figure B.1. The third `filter_gen` for the *no foil* configuration is simply an `Arm` component that does not interact with the neutron. We could insert no components for the latter but using a *mute* component can be useful to track neutrons during the simulation. We generated more files using different thickness: $50\mu m$ for the thick foil, $25\mu m$, $14\mu m$ and $12.5\mu m$ for the thin foil. In this simulation we need three different measures, like the experiment, to

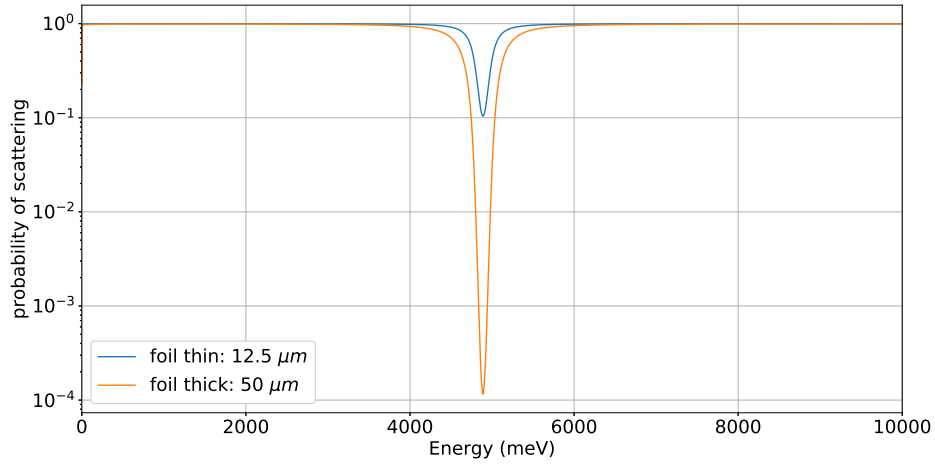


Figure B.1: The transfer functions as input of the `filter_gen` component. From JANIS database we take the cross section values

obtain enough data to compute the Compton profile. We can simulate all three runs in one execution of the code, loading 2 different thickness at the same time and splitting the

transport of the neutron-rays into three different path, not talking each other. This is achieved using the **WHEN** command and a bit of C code. After the moderator component, so after the initialization of a neutron-ray, the code choose randomly a number from $\{0,1,2\}$ and write it in the variable **foil**:

```

1 COMPONENT source = Commodus.I( Face="TS1_S02_Vesuvio.mcstas", [...])
2 AT (0,0,0) RELATIVE arm1
3 EXTEND%{
4     foil=rand()%3;
5     p*=3;
6 %}

```

This **foil** variable triggers the path choice from the **WHEN** command, the latter act like a **if** condition, placing or not placing the relative component inside the beam path.

```

1 COMPONENT filter_thin = Filter.gen(
2     filename=thin, ...)
3 WHEN (foil==1) AT (-0.158,-0.048,0.46) RELATIVE Armsample
4 GROUP Foils
5
6 COMPONENT filter_thick = Filter.gen(
7     filename=thick, ... )
8 WHEN (foil==2) AT (-0.158,-0.048,0.46) RELATIVE Armsample
9 GROUP Foils
10
11 COMPONENT No_filter = Arm()
12 WHEN (foil==0) AT (-0.158,-0.048,0.46) RELATIVE Armsample
13 GROUP Foils
14 EXTEND %{
15     SCATTER;
16 %}
17

```

After obtaining the three files (Figure B.2), the first step to extract the Compton profile is to apply the double difference method. First of all we need to set the β values:

$$\beta = \frac{\text{thin thickness}}{\text{thick thickness}} \quad (\text{B.1})$$

the thick foil is ever $50\mu m$ thick, the thin ones are 12.5, 14.0 and $25.0\mu m$ for $\beta = \{0.25, 0.28, 0.50\}$ respectively. I obtain the double difference measure applying:

$$C_{DD} = C_{out} * (1 - \beta) - C_{thin} + \beta * C_{thick} \quad (\text{B.2})$$

and we can see the results in Figure B.3.

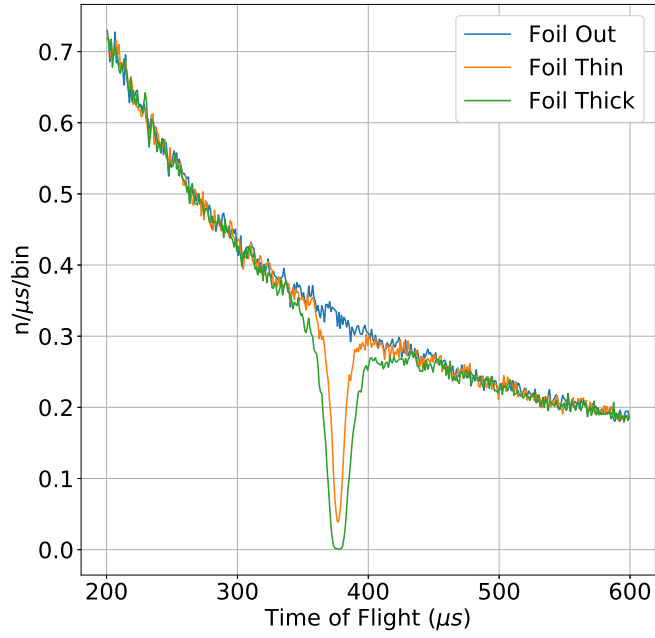


Figure B.2: The three raw data from the simulation. The thin foil as a thickness of $12.5 \mu m$ and the thick foil of $50 \mu m$ in this particular simulation.

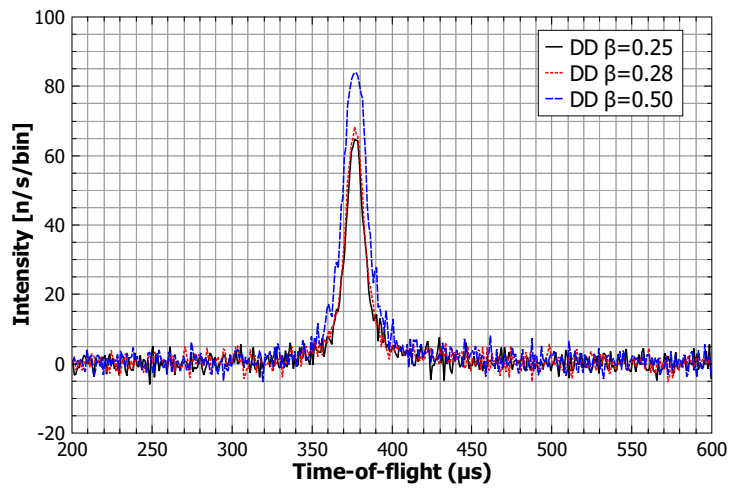


Figure B.3: In this plot we can see three measure obtained changing the thin foil thickness. The number of neutron-rays is equal for each simulation

B.1.2 Single Filter_Gen

In case of a single `Filter_Gen` component we need to precompute the total transmission function as showed in 3.11. The main improvement from the previous solution is that the number of neutron-ray reaching the detector is 3 times higher. As showed in previous chapters the Monte Carlo choice in the three foil solution split the flux in 3 parts. The p values is than corrected but the statistical error is proportional to $1/\sqrt{N}$ so

$$\text{Err}_{3f} \propto \frac{1}{\sqrt{\frac{N}{3}}} = \sqrt{3} \cdot \text{Err}_{1f} \quad (\text{B.3})$$

The other improvement is that the compiling time reduces, we need to load only one ASCII table instead of two. This method is preferable to the previous one because of the VESUVIO spectrometer background contributions are not present in simulation.

B.1.3 Generating the final energy inside the sample component

The most efficient way to simulate the resonant foil at back scattering is to include directly inside the component the resonant foil feature. On the other hand changing the foils properties become difficult and it is necessary to rewrite the component itself. This can be done using the *Cumulative Distribution Function* (CDF) obtained from the results of the double difference procedure. Assume that the function $P(x)$ is non-zero only within the region between $x = a$ and $x = b$: Define the function

$$C(x) = \int_a^x P(z) dz \quad (\text{B.4})$$

which is the probability that an event will occur with x value somewhere between a and x : By definition $P(b) = 1$ and the function will have a form something like that sketched in Fig. B.4. A random number z is generated uniformly in the interval $0 < z < 1$ and the value of x is determined as $x = C^{-1}(z)$ as sketched in Fig. B.4. This method generates values of x with the frequency that they would actually occur if the probability of an event is $P(x)dx$. This technique is already in use by the DINSMS code included into Mantid. It is actually used to correct the signal during the data reduction on the VESUVIO spectrometer. For this reason we used exactly the same CDF function, showed in figure B.4.

B.1.4 Performance Comparison

In figure B.5 we can see the double difference result obtained by the three technique explained above. All the simulation are performed with the same number of neutron-ray histories

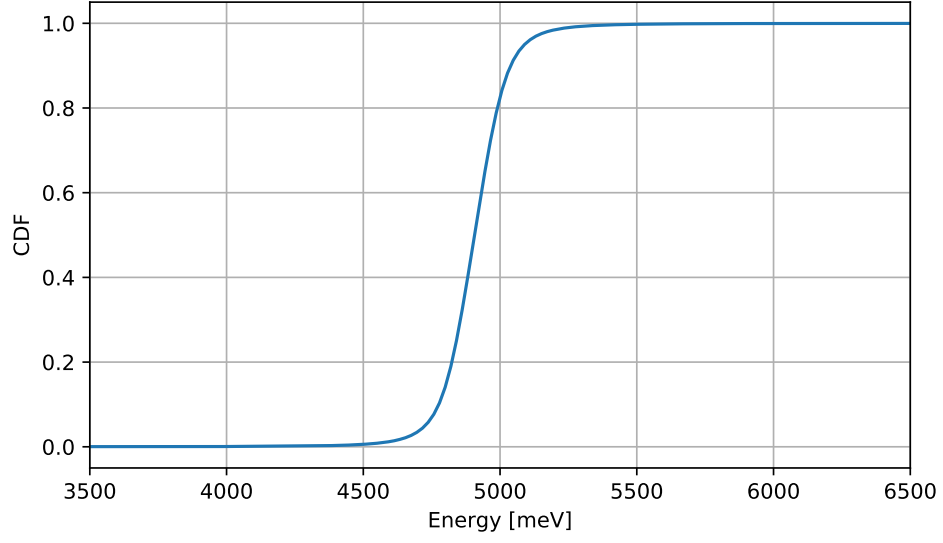


Figure B.4: Cumulative Distribution Function obtained by double difference resonant foil.

(1×10^8). The three foil technique is clearly the most inefficient cause by losing a consistent number of neutron that does not contribute to the profile intensity. The one foil technique has more resolution but in any case we are simulating a consistent number of neutron that has weight with a value near zero. The DINSMS technique is the most efficient. We are using more neutron-rays where is needed and the zero value neutron-ray are seldom generated. The difference in intensity can be explained considering the difficulty to reconstruct the foil thickness by the information taken. There are not precise measure and from bibliography research the only parameter found was β .

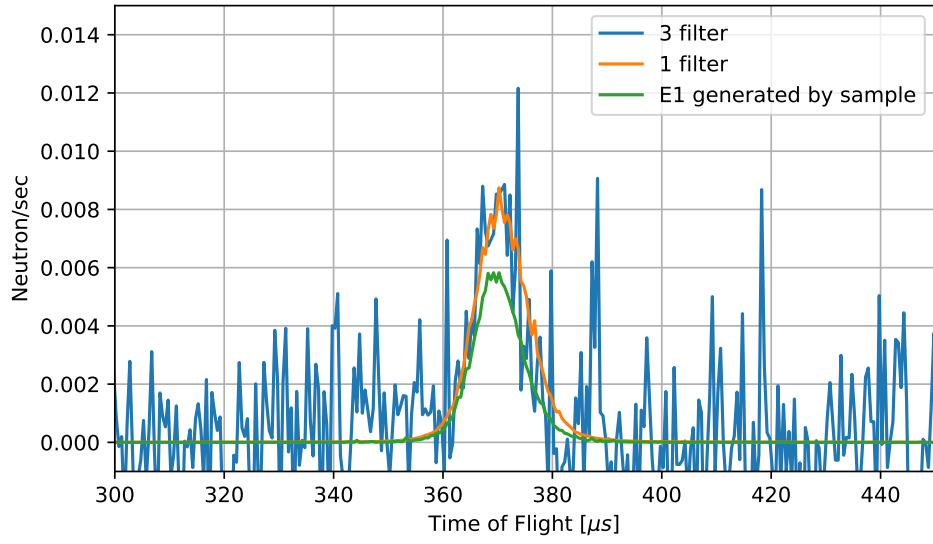


Figure B.5: Comparison between the three technique explained in this chapter. The three foil configuration (blue) has a really low resolution, the one filter configuration is quite better but using the final energy from DINSMS we notice the better resolution performance. All the simulation were taken with 1×10^8 neutron-rays.

Appendix C

Frontscattering and γ emulation

In this section we are going to investigate a simple technique that can be used to obtain a sort of gamma representation in McStas. There are not such thing as γ interaction physics inside Mcstas but, at least in concern to the VESUVIO spectrometer, we can use the same neutron-rays of the code as pseudo- γ rays. The main reason to include this part to the simulation is that front-scattering detectors are γ sensitive. The single difference technique, as explained in the introduction (see ch. 3.1), is based on a double measure obtained by inserting a second foil into the neutron path from sample to detector. The YAP detectors have been set to count the gamma emitted by the gold foil attached to them, filtering the counts by gamma energies below 600 keV, to enhance only the Gold contribution. The second foil decreases the number of neutron with energy of ~ 4.9 eV reaching the YAP.

The main problem with this method is that there are 4 cycling foil that in every period are positioned in front of 4 of the 8 banks of detectors. When the period change they move to cover the other 4 banks (refer to figure C.1). Every foil emits the gamma radiation from the neutrons incoming, so each detector counts not only the emission from his foil and the cycling one in front of it, but the cumulative contribution from all the cycling foils.

Even if the propagation time of photons (for a distance of 40-70 cm), is basically zero the detectors, mainly at large angle of scattering, are sensitive to the different positions of the foils.

In order to reproduce the experiment we use the neutron-rays from McStas adapting the configuration to simulate a gamma interaction:

- Instant propagation: we insert a detector just after every foil in the simulation
- Isotropic emission: all the detectors data collection is summed. The final result contain every gamma emitted.

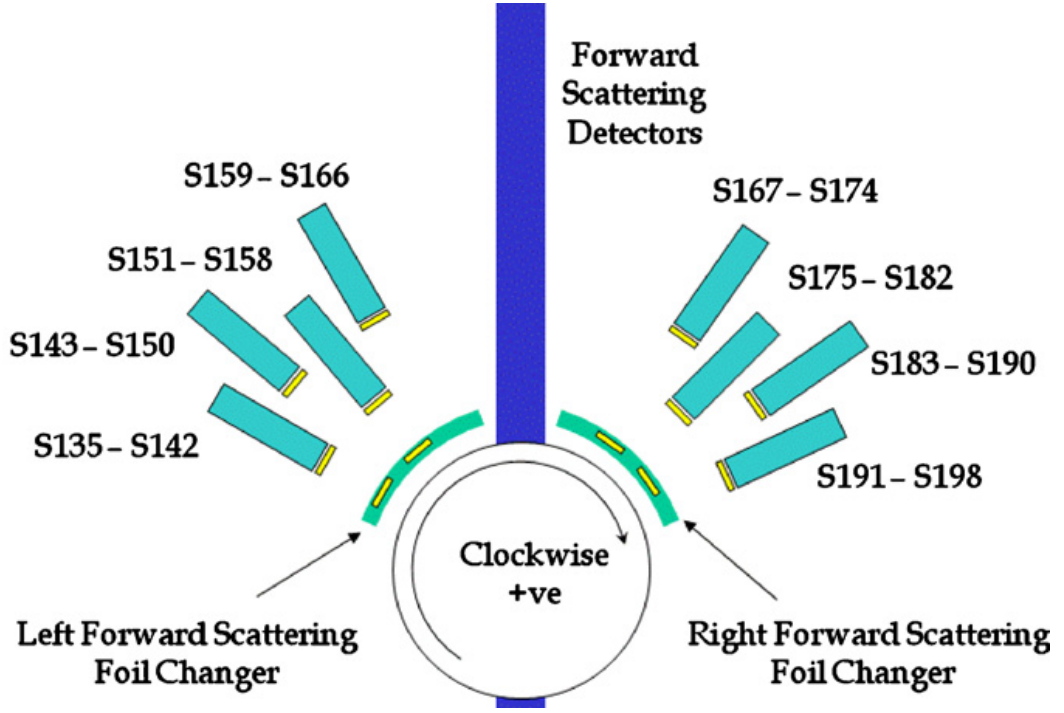


Figure C.1: The front scattering schema of the VESUVIO spectrometer. The front scattering detectors (S135-S198) are packed in group of 8, 4 above and 4 below the sample center plane.

- Photons multiplicity: We use a `filter_gen` component with a weight modifier that increase the `p` value to match the average number of photons emitted.

This gamma emission is originated in our simplified model at foils position or at *detector* position (the YAP have a foil that converts neutrons to photons). The most important approximation in this simulations is the sample emission. As we can see in fig C.2 the $n\text{-}\gamma$ cross section describe numerous resonance for incident neutron energies above 1 keV. The sample component in our case does not emit photons, we are neglecting even the de-excitation of the sample or of the gold foils.

In our case however we are going to filter neutrons at low energies where the gold show a relative isolated resonance ($\sim 4.9\text{ eV}$).

C.1 McStas Forward Model

The sequential nature of McStas induces us to consider carefully all the components that we want to insert or investigate. We are going to split neutron-rays into many paths inside the simulation because a neutron-ray can interact with only one of the cycling foils. After the sample component we make a random choice of which one foil is active. We run two

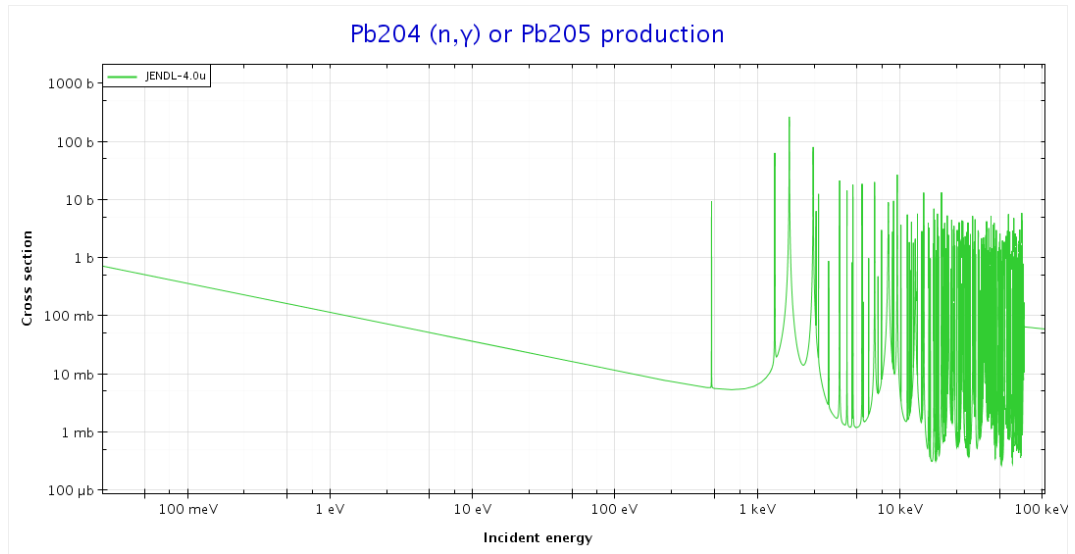


Figure C.2: From Janis Database we can see the $n\text{-}\gamma$ cross section for Lead. There are many resonance over 1 keV and this can be difficult to include the code.

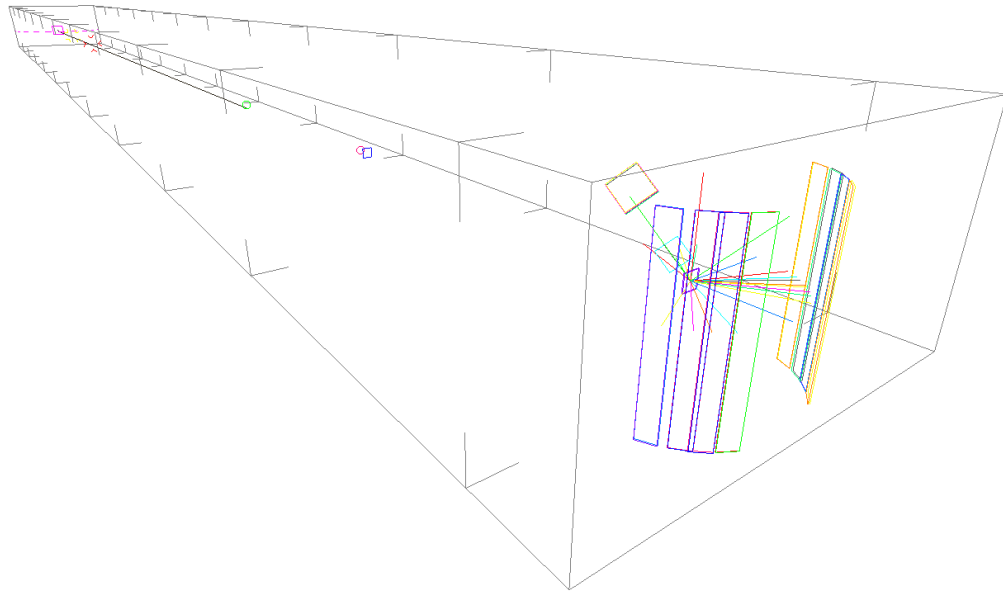


Figure C.3: The 3D visualization of the front scattering setup. The Foils are duplicated because both the foil-out and foil-in configuration are present.

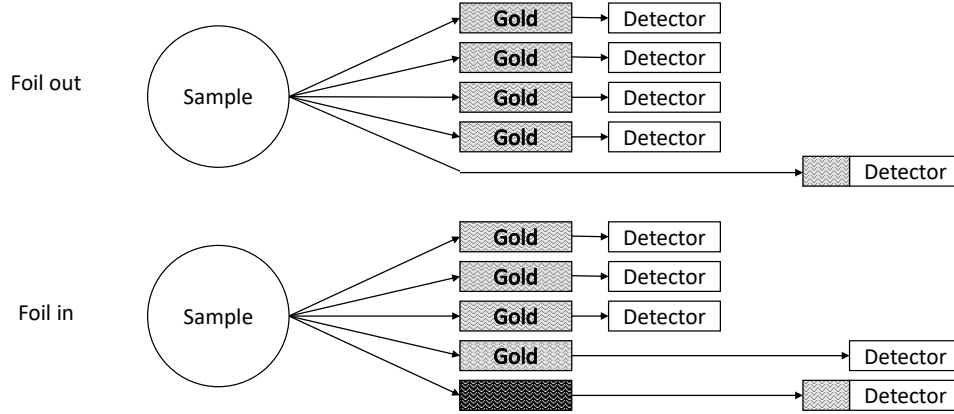


Figure C.4: The complete code, the various path choice are rearranged visually in branch. The virtual YAP are simply the union of the detector component and the filter to gamma component.

simulation for the foil in and the foil out configuration (see figure C.4). After the sample we chose randomly from 5 paths, one for every foil and one for the detector position. If we are in the configuration “foil-in” the detector path contains another `filter_gen` component to retrieve the neutrons that does not interact with the gold foil. To emulate the gold foils we use again the `filter_gen` component. In our energies range (below 1 keV) we must take into account the photons multiplicity. The conversion 1 neutron to 1 photon underrates the detector count, we can adjust this by generating a `.dat` filter file that include this. We can include the multiplicity in the cross section and calculate a new filter file. From the JANIS database we know that the photons multiplicity is constant (~ 2.8) for a neutron energy below 1 keV.

C.2 Preliminary Results

The principal assumption of the single difference technique is the good subtraction of the backgrounds gamma radiation. In our model there is not a complete background (again, there are not de-excitation or complete multiple-scattering description) but we can investigate the contribute in YAP counts by the gold cycling foils. In figure C.5 we can see the counts for each foil and detector.

We can see the the off path foils signals (from 1 to 4 in figure) are too high. The reason is that the monitor components in McStas are just plane and our setup does not reproduce the probability that a gamma is detected from the monitor. We are ignoring in this simulations

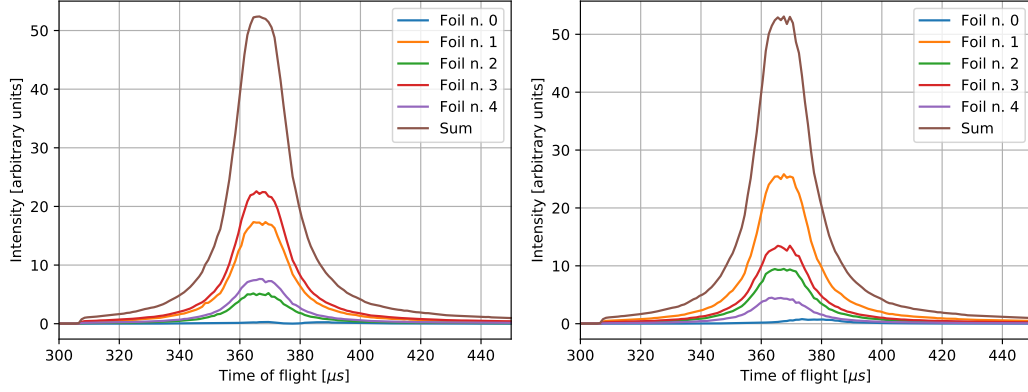


Figure C.5: In figure we can see the contribution to the signal by each foil. The foil n. 0 is the YAP foil. (left) Foil in configuration (right) Foil out configuration.

the efficiency of the YAP detector η in the VESUVIO spectrometer that can be written as

$$\eta = \frac{1}{2} \alpha \mu d \quad (\text{C.1})$$

where μ is the average attenuation length of emitted γ -rays in the scintillator glass and d is the glass thickness. The factor on half arises because half the gamma rays are emitted in the opposite direction to the detector. α is a constant ~ 1 , which can be determined by fitting to calibration data [39].

The d value at the moment can not be taken into account. A correct measure of the γ background can be obtained by McStas (as we can see in figure C.6 the oscillations on signal are presents) but we need to write down two new components:

- A detector with thickness that include the probability of interaction;
- A γ -emitting component where we can include the probability of emission.

After the development of this two component we are quite confident to reproduce and analyze each foil contribution to the signal.

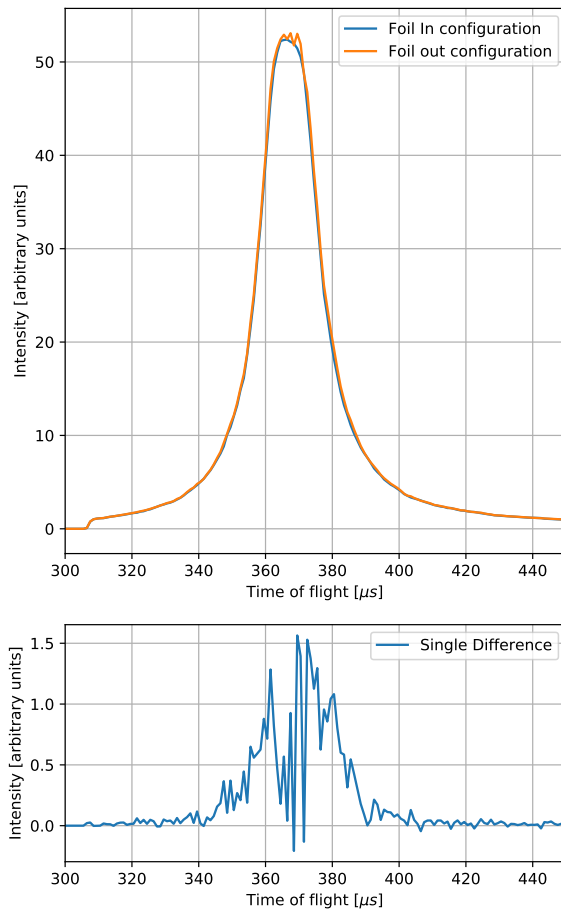


Figure C.6: The two plots are from the same simulation. In red we can see the simple subtraction from the path without interaction with the cycling foil, in black we primary sum the detector for each foil==1, foil==0 and then we subtract them.

Appendix D

Other results

During the development of the impulse approximation component we use some standard measures performed on the VESUVIO spectrometer. In these cases the sample environment is simpler because there is not a cell container, or other systems to manage temperature and pressure (CCR).

D.1 Lead

The lead sample is used by VESUVIO as calibration sample, the transfer energy is small (of the order of 200 meV so the peak position is not heavily modified by multiple-scattering contribution. The lead absorption cross section is 0.171 b while the total scattering cross section is 11.118 b. For this reason the major part ($\sim 98\%$ at 25 meV) of the neutron-rays that interact with the sample component are scattered. The lead sample is a $7\text{ cm} \times 7\text{ cm}$ square with a thickness of 0.2 cm, so the flux after the collimation is entirely included. The results are showed in figure D.1. There is a good agreement between the simulations and the multiple-scattering contribution is well estimated.

There is a small shift of the peak position to higher time of flight. The main reason of this can be a different approach to the final state effects inclusion.

The Hermite correction (eq. 7.26) can produce a negative weight, that need to be ignored in this simulation. A negative weight has non physical meaning and in the detector they can cancel out positive contributions. In our case the negative values are set to 0 and the corresponding neutron-ray is absorbed by the `ABSORB` routine.

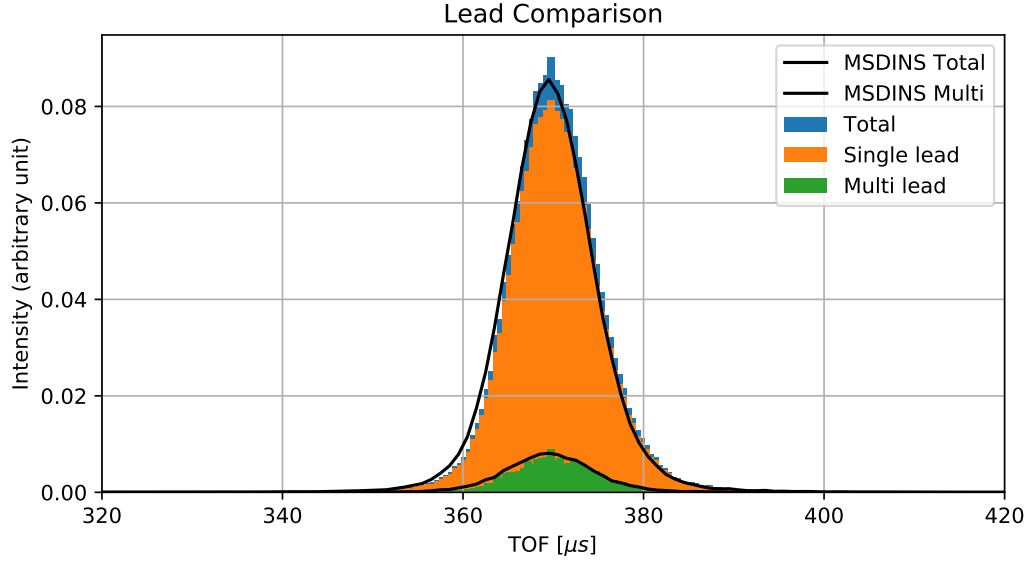


Figure D.1: The lead simulation from DINSMS code (black lines) and McStas simulation.

D.2 Vanadium

The vanadium test was performed with a disk sample with 2.5 cm radius and a thickness of 0.5 cm. In this case the absorption cross section is larger (5.08 b) and its value is near the scattering cross section (5.1 b). The results are showed in figure D.2. The discrepancies are much higher than the ones in the lead simulation. The simulation seems a bit more unstable, the high peak at $\sim 340 \mu\text{s}$ correspond to a code error that occur randomly. There are more of these peaks even at higher time of flight and it seems to be a geometry and routines lack of protection. The higher absorption cross section is mitigated by the high neutron energy: the values are referred to a 25 meV neutrons (2200 m s^{-1}) but the value that is processed by the code depends on the neutron velocity (there is a $1/v$ dependency). The higher intensity of multiple-scattering contribution can be related to a no good implementation of the absorption cross section. These results, among the D2O ones, are still under investigation.

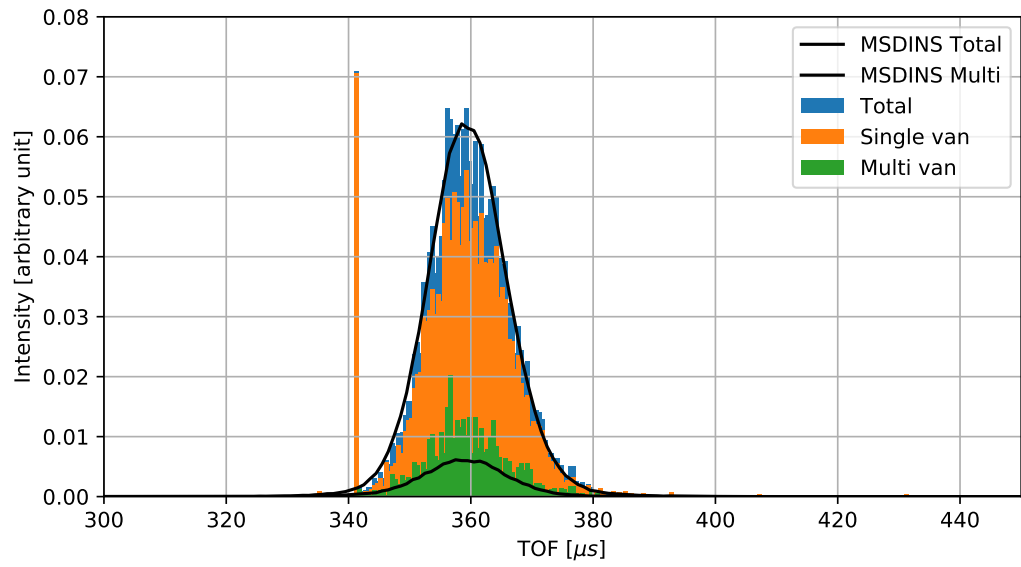


Figure D.2: The vanadium simulation from DINSMS code (black lines) and McStas simulation.

Appendix E

Variables and abbreviations

<code>ia.my_s</code>	macroscopic scattering cross section
<code>ia.my_a_v</code>	macroscopic absorption cross section (no v dependency)
<code>my_a</code>	macroscopic absorption cross section (with v dependency)
<code>my_t</code>	macroscopic total cross section
<code>ws</code>	σ_s/σ_t
<code>p_mult</code>	weight value multiplier
<code>ia.NM</code>	number of different atom masses in sample
<code>th0</code>	angle of scattering
<code>ia.xst</code>	total free scattering cross section
<code>pdcs</code>	partial differential cross section
<code>Y_west</code>	y-scaling variable
<code>ia</code>	structure for general data
<code>atom</code>	structure for single atom data
<code>IA</code>	impulse approximation
<code>NCP</code>	neutron Compton profile

Bibliography

- [1] K. Lefmann and K. Nielsen, “McStas, a general software package for neutron ray-tracing simulations,” *Neutron News*, vol. 10, pp. 20–23, jan 1999.
- [2] P. Willendrup, E. Farhi, and K. Lefmann, “McStas 1.7 - a new version of the flexible monte carlo neutron scattering package,” *Physica B: Condensed Matter*, vol. 350, pp. E735–E737, jul 2004.
- [3] P. K. Willendrup, E. B. Knudsen, E. Klinkby, T. Nielsen, E. Farhi, U. Filges, and K. Lefmann, “New developments in the McStas neutron instrument simulation package,” *Journal of Physics: Conference Series*, vol. 528, p. 012035, jul 2014.
- [4] Y. Wu, “Neutron transport theory and simulation,” in *Fusion Neutronics*, pp. 21–64, Springer Singapore, 2017.
- [5] J. Mayers, A. L. Fielding, and R. Senesi *Nuclear Instruments and Methods in Physics Research Section A*, vol. 481, p. 454, 2002.
- [6] G. McKinney, J. Durkee, J. Hendricks, M. James, and D. Pelowitz, “Mcnpx users manual, version 2.7. 0,” tech. rep., LA-CP-11-00438, “LANL, Los Alamos, 2011.
- [7] Pinna, Roberto S., Rudić, Svemir, Parker, Stewart F., Gorini, Giuseppe, and Fernandez-Alonso, Felix, “Monte carlo simulations of the toska spectrometer: Assessment of current performance and future upgrades,” *EPJ Web of Conferences*, vol. 83, p. 03013, 2015.
- [8] R. S. Pinna, S. Rudić, S. F. Parker, J. Armstrong, M. Zanetti, G. Škoro, S. P. Waller, D. Zacek, C. A. Smith, M. J. Capstick, D. J. McPhail, D. E. Pooley, G. D. Howells, G. Gorini, and F. Fernandez-Alonso, “The neutron guide upgrade of the TOSCA spectrometer,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 896, pp. 68–74, jul 2018.

- [9] M. Zanetti, S. Bellissima, L. del Rosso, F. Masi, M. Chowdhury, A. D. Bonis, L. D. Fresco, C. Scatigno, J. Armstrong, S. Rudić, S. Parker, M. Hartl, D. Colognesi, R. Senesi, C. Andreani, G. Gorini, and F. Fernandez-Alonso, “Neutronic developments on TOSCA and VESPA: Progress to date,” *Physica B: Condensed Matter*, vol. 562, pp. 107–111, jun 2019.
- [10] G. L. Squires, *Introduction to the Theory of Thermal Neutron Scattering*. Cambridge University Press, 3 ed., 2012.
- [11] C. J. Joachain, *Quantum Collision Theory: 2v.in 1v*. North-Holland, 1976.
- [12] C. Andreani, M. Krzystyniak, G. Romanelli, R. Senesi, and F. Fernandez-Alonso, “Electron-volt neutron spectroscopy: beyond fundamental systems,” *Advances in Physics*, vol. 66, no. 1, pp. 1–73, 2017.
- [13] G. B. West, “Electron scattering from atoms, nuclei and nucleons,” *Physics Reports*, vol. 18, pp. 263–323, June 1975.
- [14] E. M. Schooneveld, J. Mayers, N. J. Rhodes, A. Pietropaolo, C. Andreani, R. Senesi, G. Gorini, E. Perelli-Cippo, and M. Tardocchi, “Foil cycling technique for the VESUVIO spectrometer operating in the resonance detector configuration,” *Review Scientific Instrument*, vol. 77, p. 5103, Sept. 2006.
- [15] P. A. Seeger, A. D. Taylor, and R. M. Brugger *Nuclear Instruments and Methods in Physics Research Section A*, vol. 240, pp. 98–114, Oct. 1985.
- [16] C. Andreani, D. Colognesi, J. Mayers, G. F. Reiter, and R. Senesi, “Measurement of momentum distribution of light atoms and molecules in condensed matter systems using inelastic neutron scattering,” *Advances in Physics*, vol. 54, no. 5, pp. 377–469, 2005.
- [17] G. C. Windsor, *Pulsed Neutron Scattering*. London: Taylor & Francis Ltd., 1981.
- [18] G. Reiter and R. Silver, “Measurement of interionic potentials in solids using deep-inelastic neutron scattering,” *Physical Review Letters*, vol. 54, pp. 1047–1050, Mar 1985.
- [19] V. F. Sears, “Scaling and final-state interactions in deep-inelastic neutron scattering,” *Physical Review B*, vol. 30, pp. 44–51, July 1984.
- [20] G. Škoro, S. Lilley, and R. Bewley, “Neutronics analysis of target, moderators and reflector design for the ISIS TS-1 project,” *Physica B: Condensed Matter*, dec 2017.

- [21] “Vesuvio,” Last accessed on September 2017.
- [22] G. Romanelli, M. Krzystyniak, R. Senesi, D. Raspino, J. Boxall, D. Pooley, S. Moorby, E. Schooneveld, N. J. Rhodes, C. Andreani, and F. Fernandez-Alonso, “Characterisation of the incident beam and current diffraction capabilities on the vesuvio spectrometer,” *Measurement Science and Technology*, vol. 28, no. 9, p. 095501, 2017.
- [23] “Iris webpage,” Last accessed on November 2017.
- [24] M. Krzystyniak and *et al*, “VESUVIO+: The Current Testbed for a Next-generation Epithermal Neutron Spectrometer,” *Journal of Physics Conference Series*, in press.
- [25] G. Romanelli, A. Liscio, R. Senesi, R. Zamboni, E. Treossi, F. Liscio, G. Giambastiani, V. Palermo, F. Fernandez-Alonso, and C. Andreani, “Soft confinement of water in graphene-oxide membranes,” *Carbon*, vol. 108, pp. 199–203, 2016.
- [26] G. F. Syrykh, A. A. Stolyarov, M. Krzystyniak, G. Romanelli, and R. A. Sadykov, “Temperature dependence of the kinetic energy in the zr40be60 amorphous alloy,” *JETP Letters*, vol. 105, pp. 591–594, May 2017.
- [27] C. Andreani, G. Romanelli, and R. Senesi, “Direct measurements of quantum kinetic energy tensor in stable and metastable water near the triple point: An experimental benchmark,” *The Journal of Physical Chemistry Letters*, vol. 7, no. 12, pp. 2216–2220, 2016.
- [28] M. Krzystyniak, K. Druzicki, G. Romanelli, M. J. Gutmann, S. Rudic, S. Imberti, and F. Fernandez-Alonso, “Nuclear dynamics and phase polymorphism in solid formic acid,” *Physical Chemistry Chemical Physics*, vol. 19, pp. 9064–9074, 2017.
- [29] L. A. Rodríguez Palomino, J. Dawidowski, J. I. Márquez Damián, G. J. Cuello, G. Romanelli, and M. Krzystyniak, “Neutron total cross-section of hydrogenous and deuterated 1- and 2-propanol and n-butanol measured using the vesuvio spectrometer,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 870, pp. 84 – 89, 2017.
- [30] G. Romanelli, S. Rudic, M. Zanetti, C. Andreani, F. Fernandez-Alonso, G. Gorini, M. Krzystyniak, and G. Skoro, “Measurement of the para-hydrogen concentration in the isis moderators using neutron transmission and thermal conductivity,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. in press, 2018.

- [31] G Romanelli *et al*, “A McStas simulation of the incident neutron beam on the VESUVIO spectrometer,” *Journal of Physics Conference Series*, in press.
- [32] “Mcstas homepage,” Last accessed on February 2018.
- [33] “Scarf homepage,” Last accessed on February 2018.
- [34] Last accessed on April 2018.
- [35] S. G. Ansell S., “The viewmodisis component, mcstas simulation package (accessed on the 11 january 2018).,” 2015.
- [36] P. Willendrup, U. Filges, L. Keller, E. Farhi, and K. Lefmann, “Validation of a realistic powder sample using data from dmc at psi,” *Physica B: Condensed Matter*, vol. 385, pp. 1032–1034, 2006.
- [37] E. Farhi, V. Hugouvieux, M. Johnson, and W. Kob, “Virtual experiments: Combining realistic neutron scattering instrument and sample simulations,” *Journal of Computational Physics*, vol. 228, pp. 5251–5261, aug 2009.
- [38] R. S. Pinna, S. Rudić, M. J. Capstick, D. J. McPhail, D. E. Pooley, G. D. Howells, G. Gorini, and F. Fernandez-Alonso, “Detailed characterisation of the incident neutron beam on the TOSCA spectrometer,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 870, pp. 79–83, oct 2017.
- [39] J. Mayers and M. A. Adams, “Calibration of an electron volt neutron spectrometer,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 625, pp. 47–56, Jan. 2011.
- [40] G. Bacon, *Neutron diffraction 3 edition*. Clarendon Press.
- [41] J. Mayers, A. Fielding, and R. Senesi, “Multiple scattering in deep inelastic neutron scattering: Monte carlo simulations and experiments at the ISIS eVS inverse geometry spectrometer,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 481, pp. 454–463, apr 2002.
- [42] G. Romanelli and M. Krzystyniak, “On the line-shape analysis of compton profiles and its application to neutron scattering,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 819, pp. 84 – 88, 2016.

- [43] G. Romanelli, M. Ceriotti, D. E. Manolopoulos, C. Pantalei, R. Senesi, and C. Andreani, “Direct measurement of competing quantum effects on the kinetic energy of heavy water upon melting,” *The Journal of Physical Chemistry Letters*, vol. 4, no. 19, pp. 3251–3256, 2013.
- [44] M. Bertelsen, “Software for simulation and design of neutron scattering instrumentation.”