



Virtual experiments as a data analysis tool for neutron scattering measurements

Emmanuel Farhi

► To cite this version:

Emmanuel Farhi. Virtual experiments as a data analysis tool for neutron scattering measurements. Materials Science [cond-mat.mtrl-sci]. Grenoble 1 UJF - Université Joseph Fourier, 2014. tel-01226227

HAL Id: tel-01226227

<https://theses.hal.science/tel-01226227>

Submitted on 9 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Joseph Fourier de Grenoble

Habilitation à diriger des recherches

présentée à l'Institut Laue Langevin

Ecole doctorale: Physique

Virtual experiments as a data analysis tool for neutron scattering measurements

Emmanuel Farhi

Soutenue le 21 février 2014 à 10 heures devant le Jury composé de :

Prof. K. Andersen	Rapporteur
Dr. C. Ferrero	Rapporteur
Prof. M. Hagen	Rapporteur
Prof. K. Lefmann	Examineur (excused)
Prof. H. Schober	Examineur
Dr. J-M. Zanotti	Examineur
P. Willendrup	Examineur

This version (January 2nd 2014) is for evaluation.

Table of Contents

1 INTRODUCTION.....	3
1.1 Purpose of this manuscript.....	3
1.2 Introduction to virtual experiments.....	4
1.2.1 Definition.....	4
1.2.2 Applications.....	4
1.3 Software used in this study.....	5
1.3.1 Data analysis.....	5
1.3.2 Neutron scattering instrument simulations.....	5
1.3.3 Molecular dynamics.....	6
1.4 Modelling materials.....	6
1.4.1 From measurements.....	7
1.4.2 From molecular dynamics simulations.....	8
2 NEW TOOLS FOR VIRTUAL EXPERIMENTS.....	9
2.1 Scattering kernel for virtual experiments with isotropic density materials: the liquid Rubidium example.....	9
2.2 Data analysis and fitting tools: the iFit framework.....	13
3 TRENDS FOR VIRTUAL EXPERIMENTS.....	15
3.1 Advanced virtual experiment: the moderation of cold neutrons by liquid ^4He to produce ultra-cold neutrons.....	15
3.1.1 The dynamic structure factor of l- ^4He	16
3.1.2 The simulation model.....	18
3.1.3 Simulation results.....	19
3.1.4 UCN production rate estimates for ILL experiments.....	23
3.2 Experiment planning: the dynamics of liquid indium studied with a triple axis spectrometer.....	24
3.2.1 The liquid Indium molecular dynamics simulation.....	25
3.2.2 The structure and dynamics of l-In measured around the first sharp peak with the IN22 TAS.....	28
3.3 Data analysis with embedded virtual experiment: application to powder diffraction Rietveld refinement.....	34
3.3.1 The D2B diffractometer at the ILL.....	36
3.3.2 The powder refinement strategy.....	38
3.3.3 The refinement results.....	39
4 CONCLUSION AND OUTLOOK.....	41
4.1 Limitations of virtual experiment data analysis methodologies.....	41
4.2 What can be done to improve the applicability of virtual experiment data analysis.....	42
4.3 Looking ahead.....	43
5 References.....	44
6 Selected papers.....	45

1 INTRODUCTION

1.1 *Purpose of this manuscript*

Initially, Monte-Carlo simulations were designed to study nuclear reactions using neutron transport codes such as MCNP [1]. In the 80's, a number of 'home made' neutron scattering dedicated Monte-Carlo codes were created, e.g. NISP, MSCATT, to tackle more accurate estimates of neutron fluxes and scattering processes (e.g. multiple scattering estimates). In the late 90's, more general codes such as ResTrax [2], Vitess [3], and McStas [4][5] were initiated to allow application to a wider set of problems. These codes have first been used to design instruments by optimising their geometry and specification to reach a higher effective intensity or resolution. Then in the last decade, the notion of 'virtual experiment' – that is the tight coupling of an instrument model with a sample scattering kernel – has emerged. Even though the mechanics for this methodology is now validated, most current studies only consider the instrument design, and do not benefit from the advanced sample models now available in e.g. McStas [6][7][8].

In this manuscript, we present what can be done today using virtual experiments and how such tools may be used in a near future to analyse neutron measurements with inclusion of most data reduction and analysis steps in the experiment model. In this respect, all presented results should be considered as a prospective study, which are probably among the most advanced neutron scattering virtual experiments that can be currently envisaged.

We shall not present in details the principle of Monte-Carlo neutron scattering simulation [5][6][7][9][10], data treatment [11][12][13] and assume these topics are already known to the reader. However, a quick reminder is presented in Chapter 2, with inclusion of relevant papers at the end on this manuscript.

We shall demonstrate how we can use the McStas software to perform virtual experiments which provide invaluable information about the scattering processes in the material, including instrumental artefacts such as multiple-scattering in the sample volume, wavelength and spatial resolution, etc. This will be demonstrated in Section 3.1 with an ultra-cold neutron moderator study using a l - ^4He cell.

Then, we shall present how it is possible to plan and interpret an inelastic scattering experiment by making use of molecular dynamics to feed a virtual experiment. This will be presented in Section 3.2 with a measurement of the structure and dynamics of l -In around its first structure peak.

Finally, we shall present a novel methodology to perform Rietveld refinement of a powder structure [14], which does not make use of analytical approximation, as opposed to conventional Rietveld refinement which requires to estimate the resolution function [15]. This will be shown in Section 3.3.

For all these topics, we shall discuss the advantages and drawbacks of the Monte-Carlo methodology compared to other approaches.

1.2 Introduction to virtual experiments

1.2.1 Definition

Following [16], a neutron scattering Virtual Experiment is a simulation which:

1. models a complete instrument, including a detailed sample description,
2. provides absolute intensity results that compare with actual measurements, and
3. can be controlled like a real instrument.

The key point in these requirements is certainly the availability of advanced and accurate sample and neutron optics descriptions. We may summarize mathematically a virtual experiment as a convolution:

$$\text{Virtual experiment} = \text{Instrument simulation} \otimes \text{Advanced sample model}$$

Compared with other neutron propagation Monte-Carlo codes, the *McStas* package [4][5][6][7] has put much effort in these fields and achieved a significant breakthrough towards effective Virtual Experiments, for all classes of neutron scattering instruments and materials.

1.2.2 Applications

Once we assume that a virtual experiment is available, it may be used in many ways. The neutron facilities staff will mainly use this tool in order to upgrade or design new instruments. In this case, the simulation results will enable to estimate the main instrument characteristics, including intensity, resolution, signal to noise ratio not only on the integrated neutron beam, but also on a realistic neutron scattering signal such as diffractograms and inelastic spectra. This approach may be complemented with pure neutron transport codes such as MCNP [1] in order to study fast neutrons and gamma background *e.g.* for shielding design.

Another use of virtual experiments deals with experiment planning. An accurate virtual experiment model may be used to estimate the feasibility of a real experiment proposal, in order for instance to estimate if a scattering process in the sample is effectively measurable. An estimate of the acquisition time may also be obtained at this stage. During an experiment, one may compare the measured data with the corresponding data from the virtual experiment, in order to understand and label observed features, for instance to separate the contributions from the sample environment or parasitic scattering signals such as multiple scattering.

Last but not least, virtual experiments are invaluable for teaching. The embedded model can be shown without actually travelling to neutron facilities, and help in understanding both how instruments are built, operated and the type of data which they generate.

1.3 Software used in this study

Before we focus on the specific software used for this study, we present an overview of some available programs that may be employed for the data analysis, the modelling of neutron scattering instruments, and the molecular dynamics simulations.

1.3.1 Data analysis

Data analysis computer programs have a unique *raison d'être*: to shorten the path from a raw scientific result to publishable quality data. In order to accomplish this task, the software should be able to import, view, export, and analyse the data, including the ability to find the best scientific model to interpret it.

The usual method to analyse data from neutron scattering measurements is to follow a series of subtractions (e.g. empty sample cell), normalizations (e.g. reference sample, transmission, detector efficiency), grouping, and variable change, etc. leading to a physically meaningful data set *aka* reduced data, on which models can further be tested. These models are derived from e.g. theory and can usually be parametrized as mathematical functions. The corresponding model parameters, which are the relevant information for the scientist, are then often found by a fitting procedure.

In the following we list a number of data analysis packages, which main purpose is first to reduce (correct and transform) the raw data. The recommended packages in the scope of this manuscript are LAMP and iFit.

- **LAMP** [13] is the preferred data reduction framework used at the ILL. It can import most data files produced by neutron scattering instruments and molecular dynamics modelling software, and provide a set of reduction routines. It has been used to pre-process the liquid helium data in section 3.1.
- **iFit** [11][12] is a generic light-weight data analysis package for Matlab, which can import any data file, apply mathematical operators, visualise, export, fit and optimise multi-dimensional problems. Most of the data analysis and the figures of this manuscript have been obtained with iFit.
- **FullProf** [17] is a powerful software for powder diffraction, Fortran based, which implements the Rietveld method [14]. This method consists in a parametrisation of the instrument resolution function [15] and a fit of a physical model using the raw diffractogram with most reduction steps included analytically in the model. It is based on the Crystallographic Fortran Module Library (CrysFML). FullProf and CrysFML are used in the section 3.3.

1.3.2 Neutron scattering instrument simulations

Many Monte-Carlo based software have been written to model neutron scattering instruments. An instrument description, which consists in geometrical and physical parameters, is given as input. The result of the modelling is a set of neutron events and distributions. The most widely spread package is currently McStas.

- **McStas** [4][5][9] is a general neutron instrument Monte-Carlo simulation software used by a wide community. All classes of neutron instruments, including polarized ones, are supported. Computation is C based, with a separate graphical layer. The software supports multi-processing and clusters, parameter optimisation, instrument geometry and provides plotting capability of simulation results. It includes extensive documentation and examples, a large instrument component library and material data base. Significant effort has been put into virtual experiments with the developments of numerous sample kernels to model all types of neutron/matter interaction for many materials. All simulations presented in this manuscript have been obtained with McStas.

1.3.3 Molecular dynamics

Molecular dynamics (MD) consists in the modelling of atom and molecule movements, as a function of time, in a simulation box. The result is usually a trajectory file which should be analysed subsequently. In the scope of this manuscript, we have used VASP and nMoldyn.

- **VASP** [18][19][20] is an *ab-initio* plane wave based DFT code for periodic systems. Its main applications are to perform electronic calculations and quantum mechanical molecular dynamics from first principle. Such simulations have few tunable parameters (energy thresholds, pseudo-potential choice, approximation/speed/precision balance).
- **nMoldyn** [21][22][23] is a molecular dynamics trajectory analysis software, Python based, which can import trajectories from most MD codes such as CASTEP, VASP, NAMD and DL-POLY. Coordinate analysis in real and reciprocal space can be performed to extract structural and dynamical informations. In this manuscript, we have used nMoldyn to compute the dynamic structure factor $S(q, \omega)$ from molecular dynamics simulations.

1.4 Modelling materials

Virtual experiments require that the materials in the neutron beam be known with structural, and possibly dynamical information.

In principle, once the measured or simulated data has been corrected and transformed to the momentum/energy reciprocal space, it can be used as a sample model for virtual experiments. In practice, the main sample kernels supported by McStas are powders, single crystals and all isotropic materials (liquid, gas, powder, amorphous material), as seen in Table 1.

Other sample models can be used but are much more restrictive as they are analytical. This is for instance the case of small angle scattering contribution which is modelled using analytical form and structure factors. In such cases, this information will be obtained from a fit of sample parameters to the simulated or measured data.

The purpose of the following sections is to detail how the information needed as input to these sample kernels can be obtained, both from previous experiments and molecular dynamics simulations.

<i>Sample kernel</i>	<i>Scattering</i>	<i>Material</i>	<i>Parameters</i>
<i>Single_crystal</i>	Coherent and incoherent, elastic	Single crystals	Lattice, $ F^2(hkl) $, σ_{inc} and σ_{abs}
<i>PowderN</i>	Coherent and incoherent, elastic	Powders	$ F^2(hkl) $, σ_{inc} and σ_{abs}
<i>Isotropic_Sqw</i>	Coherent and incoherent, elastic	Powders, liquids, gas, amorphous (polymers, glass)	$ F^2(hkl) $ or $S(q)$, σ_{inc} and σ_{abs}
<i>Isotropic_Sqw</i>	Coherent and incoherent, elastic and inelastic	As above	$S_{coh}(q, \omega)$, $S_{inc}(q, \omega)$, σ_{abs} , σ_{inc} and σ_{coh}
<i>SANS kernels</i>	Coherent and incoherent, elastic	Large scale structures (polymers, solutions, ...)	Parameters of the form factor $P(q)$ and the structure factor $S(q)$, σ_{inc} and σ_{abs}

Table 1: Main sample kernels used to model materials with McStas.

1.4.1 From measurements

Data acquired from existing measurements (time-of-flight, triple-axis, small angle, diffractometer, ...) require to carefully correct the raw data to remove any contribution that does not originate from the sample itself. This includes for instance the following contributions:

- the instrument resolution (in momentum and energy),
- the sample environment scattering (cryostat, furnace, ...),
- the sample container scattering and shape (usually flat or cylindrical cell),
- any additional scattering on materials in the beam other than the sample, for instance parasitic scattering – phonons and multiple-scattering – on the instrument optics components such as filters or monochromator mounts,
- the neutron attenuation correction through the sample, known as the secondary extinction or self shielding,
- the ambient background around the instrument,
- the detector efficiency and parallax effect,
- the measurement normalisation to a reference sample (e.g. vanadium) which mostly corrects for the detector efficiency on the elastic scattering,
- multiple orders scattering, that is a non ideal monochromaticity of the beam,

- the multiple-scattering in the sample.

Except for the multiple-scattering contribution and the instrument resolution function, all of the above corrections can be performed using linear combinations, usually weighted with the transmission through the sample and neutron energy dependent factors. The triple-axis spectrometer measurements data analysis often take into account a full 4-dimensional resolution model, but most other neutron scattering measurements only consider one dimensional Gaussian functions. The data reduction can be performed using software such as LAMP [13], Mantid [24], DAVE [25], and GRASP [26].

After being corrected and reduced, it is usual to transform the data measurement axes (position, angle, time) into momentum and energy axes, which are used to characterize the sample structure and dynamics. This transformation is achieved by applying a Jacobian. In some cases, such as in the Rietveld refinement method [14] for diffraction measurements, part of the data reduction, the resolution function corrections and the angle to momentum transformation are included in the fitting procedure which provides the final physical parameters. The reader may refer to [27] for a detailed presentation of such corrections and transformations.

Once the data has been treated, structural and dynamic information can be extracted: list of lattice reflections $|F^2|$, structure factor $S(q)$, dynamic structure factor $S(q, \omega)$, form factor $P(q)$, and any other information to be used as input. For instance, it is easy to read CIF entries from a past structural experiment and obtain the list of lattice reflections $|F^2|$. Similarly, most neutron inelastic scattering time-of-flight measurement can be converted into a dynamic structure factor $S(q, \omega)$.

1.4.2 From molecular dynamics simulations

Molecular dynamics simulation can provide extensive structural and dynamic information when existing measurements are not available or satisfactory. A set of atoms and molecules are positioned in a simulation box. The total energy of the system provides the forces that act between the atoms and molecules. The equations of motion are then integrated, providing trajectories over time. This manuscript will certainly not present the complex underlying theory in classical and *ab-initio* molecular dynamics codes and we rather encourage the reader to refer to [28] for these aspects. In the following, we shall simply assume that a trajectory can be obtained for a given material. Many structural and dynamical analysis can be further performed:

- mean square displacement in order to extract the diffusion coefficient D ,
- radial density function $g(r)$, measuring the probability to find neighbours at a given distance from a central atom,
- structure factor $S(q)$, which is basically the Fourier transform of $g(r)$,
- classical (symmetric) dynamic structure factor $S(q, \omega)$, which provides both structural and dynamical information, including for instance coherent scattering from phonons and incoherent scattering from disorder.

The dynamic structure factor contains all of this information, and is sufficient to describe most of the material features measured with neutrons.

In practise, we have extensively used the *ab-initio* DFT code VASP [18] to model e.g. metallic liquid systems, and then computed $S(q, \omega)$ with nMoldyn [21][22][23]. When the inelastic scattering part is not required, the structure factor $S(q)$ may be computed faster to model e.g. liquid and powder structures.

2 NEW TOOLS FOR VIRTUAL EXPERIMENTS

In this section, we present recent achievements which may constitute parts of a data analysis procedure using virtual experiments. These tools have been developed within the last decade, are published, and freely available for the community [7][9][11][12].

2.1 Scattering kernel for virtual experiments with isotropic density materials: the liquid Rubidium example

The scattering and moderation of neutrons by an isotropic material can be modelled using the McStas *Isotropic_Sqw* component [4][5][9]. In this component, we consider an isotropic medium illuminated with thermal or cold neutrons. We ignore quantum effects, refraction and primary extinction, nuclear processes except neutron absorption, and assume no Breit–Wigner resonance in the incoming neutron energy range. Following Squires [29], we relate the neutron double differential scattering cross section, that is the measured scattering intensity, to the material total scattering law $S(q, \omega)$

$$\frac{d^2 \sigma}{d\Omega dE_f} = \frac{\sigma}{4\pi} \frac{k_f}{k_i} N S(q, \omega) .$$

The double differential cross section describes the number of neutrons scattered per unit solid angle $d\Omega$ and per unit final energy dE_f . In this equation, σ is the bound atom scattering cross section [30], E_f , E_i , k_f , k_i are the kinetic energies and wavevectors of final and initial states respectively, $N = \rho V$ is the number of atoms in the scattering volume V with atomic number density ρ , Ω is the solid angle and q , ω are the wavevector and energy transfer at the sample. In practice, the double differential cross section is a linear combination of the coherent and incoherent parts of the dynamic structure factor with weighting given by the coherent and incoherent bound scattering cross sections σ_{coh} and σ_{inc} respectively so that $\sigma S = \sigma_{coh} S_{coh} + \sigma_{inc} S_{inc}$.

Following Sears [31], we evaluate the total scattering cross section per scattering unit as a function of the incident neutron energy by integrating the differential scattering cross section over the full space solid angle and all neutron final energies

$$\sigma_s(E_i) = \frac{\sigma}{4\pi} \iint \frac{k_f}{k_i} S(q, \omega) d\Omega dE_f .$$

Since $\Omega = 2\pi(1 - \cos \theta)$ with θ being the scattering angle, $\omega = E_i - E_f$ and $\vec{q} = \vec{k}_i - \vec{k}_f$ we draw

$$\sigma_s(E_i) = \frac{\sigma}{2k_i^2} \int_{-\infty}^{E_i/\hbar} \int_{q_{min}(\omega)}^{q_{max}(\omega)} q S(q, \omega) dq d\omega$$

where we have defined the wavevector q integration range between

$q_{min}(\omega) = |\vec{k}_i - \vec{k}_f| = \sqrt{2m_n/\hbar^2} \sqrt{E_i - \sqrt{E_i - \hbar\omega}}$ and $q_{max}(\omega) = |\vec{k}_i + \vec{k}_f| = \sqrt{2m_n/\hbar^2} \sqrt{E_i + \sqrt{E_i - \hbar\omega}}$ with m_n being the neutron mass. The energy dependent total cross section is then computed as

$$\sigma_{tot}(E_i) = \frac{\sigma_{abs}^{2200m/s}}{\sqrt{2E_i/m_n}} + \sigma_s(E_i)$$

where $\sigma_{abs}^{2200m/s}$ is the absorption cross section for neutrons with velocity 2200 m/s [5] and we assume no absorption resonance. This quantity allows to determine the total scattered intensity, as described below.

Obtaining the dynamic structure factor

The dynamic structure factor is usually obtained from a past experiment or a molecular dynamics simulation, as detailed in section 1.4. It may also be derived from an analytical model. When obtained from a previous measurement, e.g. on a neutron time-of-flight spectrometer, a number of data corrections must be applied in order to extract the scattering law: parallax corrections in case of complex non-spherical detector geometries, normalisation to a reference sample (e.g. Vanadium rod), detector efficiency, k_f/k_i ratio, empty cell and cadmium masks subtraction, time to energy transfer and angle to momentum transfer variable changes. In the case of a molecular dynamics simulation, the computed dynamic structure factor is the so-called classical limit scattering law $\tilde{S}(q, \omega) = e^{-\hbar\omega/2k_B T} S(q, \omega)$ which is symmetric in energy. At high temperatures compared to the material dynamics energies, the dynamic structure factor coincides with its classical limit.

The condition of detailed balance rules the material modes population at temperature T so that $S(q, -\omega) = e^{-\hbar\omega/k_B T} S(q, \omega)$. This expression can be used to symmetrise the energy range. The resulting scattering law S can then be defined over a momentum range $[0, Q_{max}]$ and energy range $[-\omega_{max}, \omega_{max}]$.

Normalisation and moments of the scattering law

The static structure factor can be simply estimated from the scattering law

$$S(q) = \int_{-\infty}^{\infty} S(q, \omega) d\omega \simeq \int_{-\omega_{max}}^{\omega_{max}} S(q, \omega) d\omega$$

where we implicitly assume that most of the scattering takes place around the elastic line condition $\omega=0$, and that the energy transfer range $[\omega_{max}, \omega_{max}]$ contribution is negligible [32].

This structure factor $S(q)$, both for coherent and incoherent parts, converges to 1 when $q \rightarrow 0$. An other normalisation rule for the coherent part, derived from the pair distribution function $g(r)$ for $r=0$, can be used to check the absolute value of $S(q)$ [33]

$$\int_0^{\infty} q^2 [S(q) - 1] dq \simeq \int_0^{Q_{max}} q^2 [S(q) - 1] dq = -2\pi^2 \rho$$

The moments of the coherent scattering law can be computed in order to estimate the collective excitation and harmonic frequencies [34], as detailed in section 3.2.1. In the following, we define the norm of S on the available dynamic range as

$$|S| = \int_{-\omega_{max}}^{\omega_{max}} \int_0^{Q_{max}} S(q, \omega) dq d\omega$$

Penetration depth and the choice of the energy and momentum transfer

Neutrons entering the material can be transmitted, scattered or absorbed. The transmission probability for a penetration depth d is $\exp^{-\rho \sigma_{tot}(E_i) d}$. The scattered events represent a fraction

$\sigma_s(E_i)/\sigma_{tot}(E_i)$ of the non transmitted neutrons. In view to properly take into account the secondary absorption in the material, we cast a random number ξ_d in the range [0,1] and determine a penetration depth along the path as

$$d_s(E_i) = \frac{\ln(1 - \xi_d [1 - e^{-\rho \sigma_{tot}(E_i) d_{tot}}])}{\rho \sigma_{tot}(E_i)}$$

where d_{tot} is the full path length in the material volume.

In order to select an energy and momentum transfer, we use the scattering law S , assuming it is a probability density function of the continuous uniform variables q and ω , describing possible scattering events in the material. We could select randomly a momentum and energy transfer and weight the scattering event with the probability $S(q, \omega)/|S|$. But in order to orient the choice of the energy and the momentum transfer towards regions of higher probability in the scattering law (that is e.g. elastic line, Bragg peaks, phonons, ...), we apply a variance reduction by choosing the energy and momentum on probability distributions derived from projections of the dynamic structure factor.

To choose the energy transfer, we build the probability distribution of energy $P_\omega(\omega)d\omega$

$$P_\omega(\omega) = \frac{1}{|S|} \int_0^{Q_{max}} q S(q, \omega) dq .$$

We choose a random number ξ_ω in the range [0,1] and determine the energy transfer ω from the primitive of P_ω so that

$$\xi_\omega = \int_{-\omega_{max}}^{\omega} P_\omega(\omega') d\omega' .$$

Similarly, we define the probability distribution of momentum $P_q(q|\omega)dq$, for the previously selected energy transfer ω , as

$$P_q(q|\omega) = q S(q, \omega) / S(q) ,$$

and we select a random number ξ_q in the range [0,1] to determine the momentum transfer q from the primitive of P_q so that

$$\xi_q = \int_0^q P_q(q'|\omega) dq' .$$

Finally, the conservation laws (selection rules)

$$\omega = E_i - E_f \quad \text{and} \quad \vec{q} = \vec{k}_i - \vec{k}_f$$

are checked and the scattering cone is determined with an angle θ w.r.t. the incoming neutron direction

$$\cos \theta = (k_i^2 + k_f^2 - q^2) / 2k_i k_f .$$

When these conditions can not be verified, namely $|\cos \theta| > 1$, a new choice for the energy and the momentum is made. In the end, the events that can be scattered describe the dynamical range implied by the selection rules. When the scattering event can be defined, a new penetration depth is determined and the procedure is iterated until the neutron event exits the material, to account for multiple scattering.

It must be stressed that in order to properly estimate multiple scattering events for an incident neutron wavevector k_i , the dynamic structure factor must be known up to $q_{max}(\omega=0) = 2k_i$. This implies that it is not possible to extract the scattering law from an experiment in view to estimate the included multiple scattering contribution. The estimate would then only be valid up to $q = k_i/2$.

Benchmarking the *Isotropic_Sqw* component against experiments: liquid Rubidium

The *Isotropic_Sqw* component for McStas can be used to model the scattering of neutrons in sample materials, inserted in a realistic instrument description. The data produced by such a model resembles the one gathered at a real instrument. It does for instance include resolution effects, neutron multiple-scattering, scattering in the sample environment, self shielding, and geometry effects.

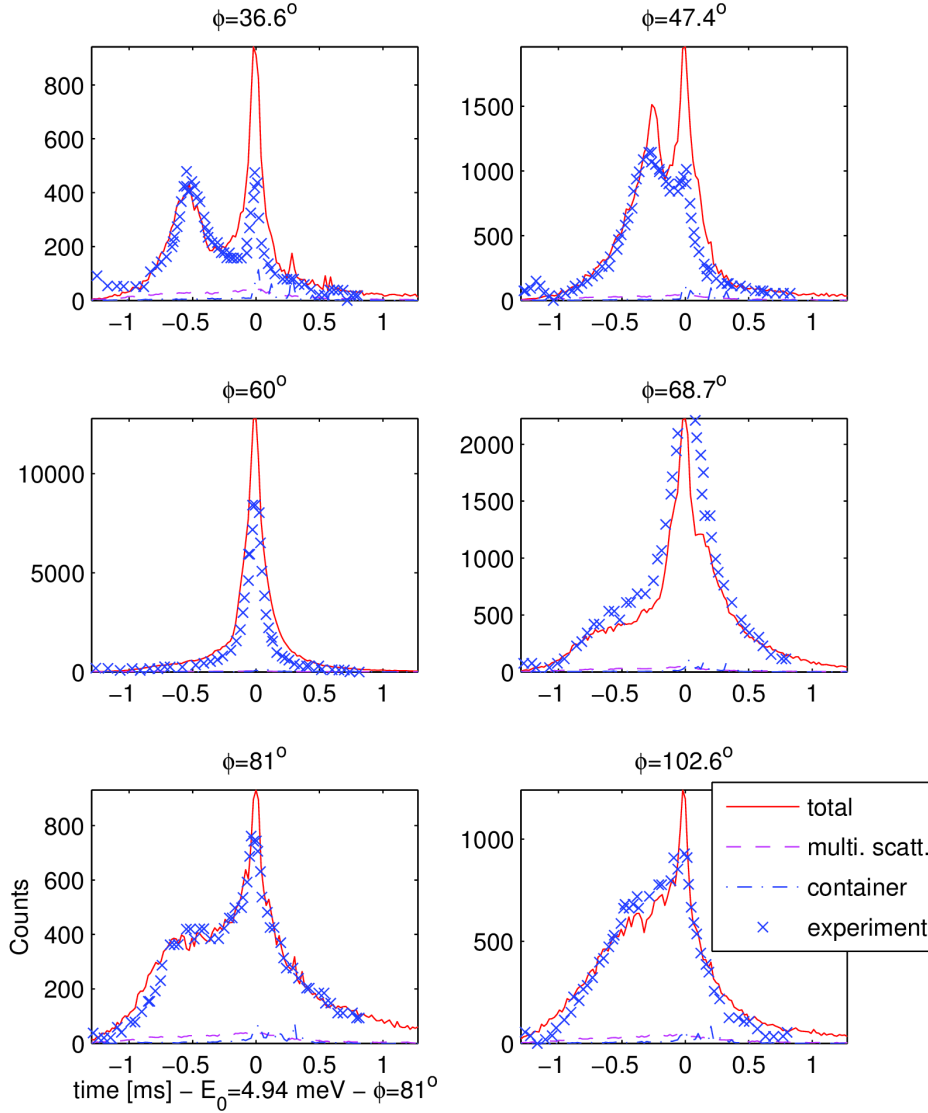


Illustration 1: The liquid rubidium neutron scattering (raw data) with incoming flux at $E_0 = 4.94$ meV showing the simulated total signal from sample (line) compared with the experimental data (crosses) from Ref. [35]. The sample environment (dash-dotted line, low intensity) and multiple scattering (dashed line) simulated contribution are also shown. Data and figure from [7].

A validation procedure for the component would require to treat the simulated data the same way as a real experiment. However, this methodology introduces many treatment artefacts associated to the approximations implied by the treatment steps.

When the raw data and the exact instrument configuration are known, it is possible to directly compare the simulated data set with the measurements. Unfortunately, few published work include such raw data in parallel with the treated data. We have identified two such works published previously: a liquid caesium [36] and a liquid rubidium [35] neutron inelastic scattering study. We have chosen to validate the *Isotropic_Sqw* scattering kernel on the rubidium experiment, as it provides a detailed experimental configuration, as well as extensive raw measurement data.

The Illustration 1, extracted from [7], presents the liquid rubidium raw data measured by Copley [35], as well as the simulated scattering as a function of acquisition time, for a set of scattering angles, with an incoming cold neutron energy of 4.94 meV. The simulated data set was obtained with a simple time-of-flight instrument description using McStas, and a dynamic structure factor computed from a classical molecular dynamics with 512 Rb atoms above the melting point.

A copy of the publication validating the McStas *Isotropic_Sqw* component [7], as well as the comparison of the simulation with the experimental results and the MSCAT code is available at the end of this manuscript (page 45).

2.2 Data analysis and fitting tools: the iFit framework

In order to be able to treat data sets from many different sources, seamlessly, we have developed a data treatment framework named iFit [11][12]. The LAMP software [13] could have been used, but its internal object definition (workspace) does not allow to retain the full content of any data source, does not provide data agnostic mathematical operators, and its fitting/optimization capabilities are more limited. As detailed below, iFit goes beyond usual numerical analysis frameworks such as bare NumPy, Matlab, and IDL, as it propagates metadata information (axes, error bars, links, history, monitor, ...) thorough all available class methods.

- **Data objects:** The *iData* class is a container for any multi-dimensional data set. It has a limited number of static properties, and a flexible storage area to hold e.g. the file content structure. In addition, the class supports internal links/shortcuts, which allow to reach any part of the object, including any specific portion of the storage area and even external resources (files, distant content), with a user defined name. The class definition provides a large number of methods which can apply to any data set. These methods include many mathematical operators. The *Signal* and *Axes* links are always defined, and allow e.g. to properly handle binary operators, especially when the operands do not perfectly overlap, in which case a re-sampling is automatically performed. The *iData* objects can be exported to many different file formats.
- **Model objects:** The *iFunc* class can hold a semi-symbolic representation of any multi-parameter model, of any dimensionality. A mathematical expression is used for the evaluation of the model. Just as for the data class above, a large number of methods allow to combine any sequence of data and model objects together, to create more complex models. One important method is the fit to a data set, which will find optimal parameters to match a

given data set, for any dimensionality. The fit routine may use any of the 23 optimisers provided with the framework.

- **Data importation:** the iFit framework can currently import about 58 different data formats. The detection of the format is automatic, and the resulting *iData* data object is standardised, still retaining the full content of the initial file.
- **Libraries:** The framework is distributed with a set of file format readers, and many optimisers. These libraries can be used independently of iFit for inclusion into other projects. Predefined models are available to create more complex iFunc objects.

For instance, using iFit, a data file, whatever be its content, is imported into an iData object with:

```
>> a = iData( file )
```

then the object can be plotted, whatever be its dimensionality, with:

```
>> plot(a)
```

An extensive set of mathematical methods can be applied to the data set objects, including trigonometry, exponentiation and logarithm, convolution, FFT, wavelets, binary operators, statistics, derivatives and integrals, interpolation, slicing and appending.

Multi-dimensional models can be defined with a semi-symbolic mechanism. For instance,

```
>> f = iFunc( 'p(1)*x+p(2)' )
```

defines a 1D line. A set of 39 predefined models is provided (Gaussian, Lorentzian, polynomials, ...). The model objects can be manipulated and combined with a large set of mathematical operators. A Voigt function is simply defined as the convolution of a Gaussian and a Lorentzian:

```
>> v = convn(gauss, lorz)
```

A squared Gaussian is defined with the element-wise multiplication ($\cdot$ *) or exponentiation ($\cdot$ ^), and a 2D Gaussian is defined with the matrix orthogonal exponentiation operator (^)

```
>> gauss_squared = gauss.^2
```

```
>> gauss_2d = gauss^2
```

Once model and data set objects have been instantiated, they can be used to determine the best model parameters to match the data set:

```
>> p = fits(data, model)
```

A publication presenting the iFit framework [11] is available at the end of this manuscript.

3 TRENDS FOR VIRTUAL EXPERIMENTS

Once neutron scattering instruments can be simulated, samples can be modelled and optimisation procedures are available, it is possible to envisage a number of applications to virtual experiments. Indeed, a virtual experiment produces data which can directly be compared with the actual experiments. Moreover, the virtual experiment can be used as a model, which, compared to measurement data, allow to refine instrument and sample parameters.

As a start, we shall present an advanced virtual experiment of a ultra-cold moderator, which reproduces experimental data, and allows to estimate the ultra-cold neutron production rate at various beam lines.

The first application of virtual experiment data analysis is to optimise the instrument configuration, in order to maximise the beam time allocation outcome. The criteria for the optimisation is then a feature extracted from the simulated detector signal, figuring for instance phonon or Bragg peak intensities and widths. This allows for instance to support an experiment proposal or a new instrument configuration design. In principle, this methodology can be used with McStas, but can be generalised to other simulation codes such as Vitess [3] and MCNP/X [1].

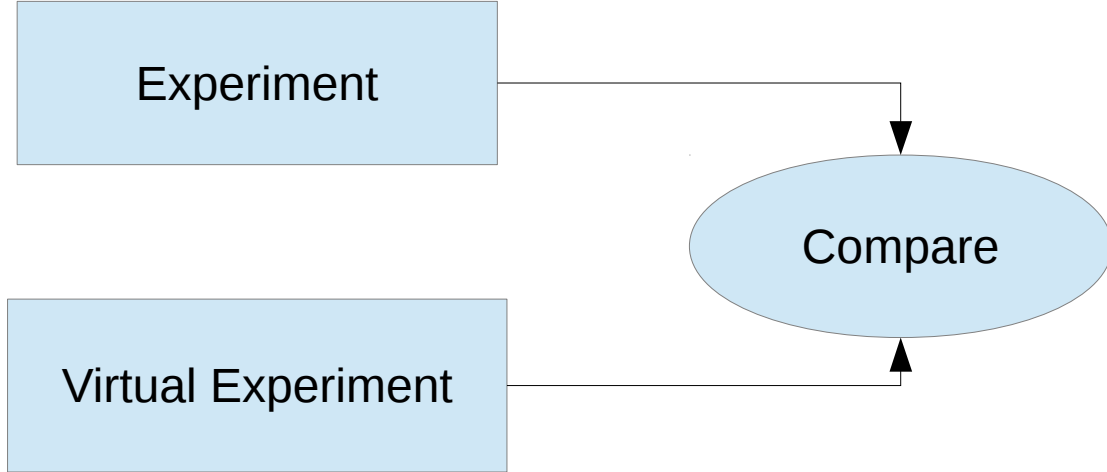
An other application is the analysis of an experiment result without explicit reduction step. In this way, the real raw measurement is compared directly with the corresponding virtual experiment. The sample should then include an analytical model inferred from e.g. theory, usually in reciprocal space and energy. Then, by tuning the model parameters, the simulated data set is matched with the experiment results in an optimisation loop. We shall demonstrate this approach in the scope of a powder refinement.

All topics presented below are original work, currently unpublished, and should be considered as potential paths for data analysis based on virtual experiments.

3.1 Advanced virtual experiment: the moderation of cold neutrons by liquid ^4He to produce ultra-cold neutrons

In this section, we show how a virtual experiment can be used to understand a past experiment, get more insight on the moderation mechanism, and forecast expected capabilities of existing and planned experimental set-ups. The schematic representation of this section methodology is shown as Drawing 1.

Ultra cold neutrons (UCN) are commonly used for fundamental physics studies, such as the determination of the neutron life-time, its electric dipole moment, or the interaction of the neutron in the gravitational field. Currently, the best UCN production facility is located at the ILL, and uses the neutron turbine PF2 (slowing down neutrons by Doppler effect). In view to ease the access to UCN's, it has been envisaged to produce slow neutrons by transferring the kinetic energy of incoming 1 meV neutrons to the liquid ^4He phonon branch.



Drawing 1: Understanding a past experiment by comparison with a virtual experiment.

In the following, we focus on the simulation of a liquid helium cell illuminated with a neutron beam from a velocity selector.

3.1.1 The dynamic structure factor of l - ^4He

We have used the dynamic structure factor measured from 3 past experiments [37][38][39], reduced with LAMP [13]. In all cases, the experimental data is corrected as described in section 1.4.1 above, and the resulting $S(q)$ is scaled to match the structure factors maximum value measured by Robkoff and Hallock [40] at 1.7K. The Illustration 2 presents the structure factor $S(q)$ obtained from experimental data [37] as a function of pressure, at 1.7 K. However, we emphasise a strong dispersion in the previously published $S(q)$ data as a function of pressure [41][42].

The full scattering law is shown in Illustration 3 at SVP and 25 bar. The incoming free neutron dispersion $\omega = \hbar^2 q^2 / 2m$ is also indicated. The intersection with the single phonon (elementary excitation) dispersion occurs for $\omega \sim 1$ meV and $q \sim 0.7 \text{ \AA}^{-1}$.

Compared to SVP, the high pressure $S(q, \omega)$ displays a steeper low- q acoustic dispersion of the elementary excitation, as well as a weaker intensity, as seen on Illustration 2 for momentum transfer values q below the roton maximum. The roton intensity is increased with pressure.

Such $S(q, \omega)$ data sets have been produced for l - ^4He at pressure SVP, 5, 10, 15, 20 and 25 bar.

Integrating the differential scattering cross section as a function of the incoming neutron energy, we compute the total scattering cross section shown in Illustration 4. These cross sections for p =SVP, 5, 10, 15, and 20 bar and 1 meV neutrons are found to be 19.3, 11.4, 6.6, 4.2 and 3.6 mbarn respectively with the [39] data set.

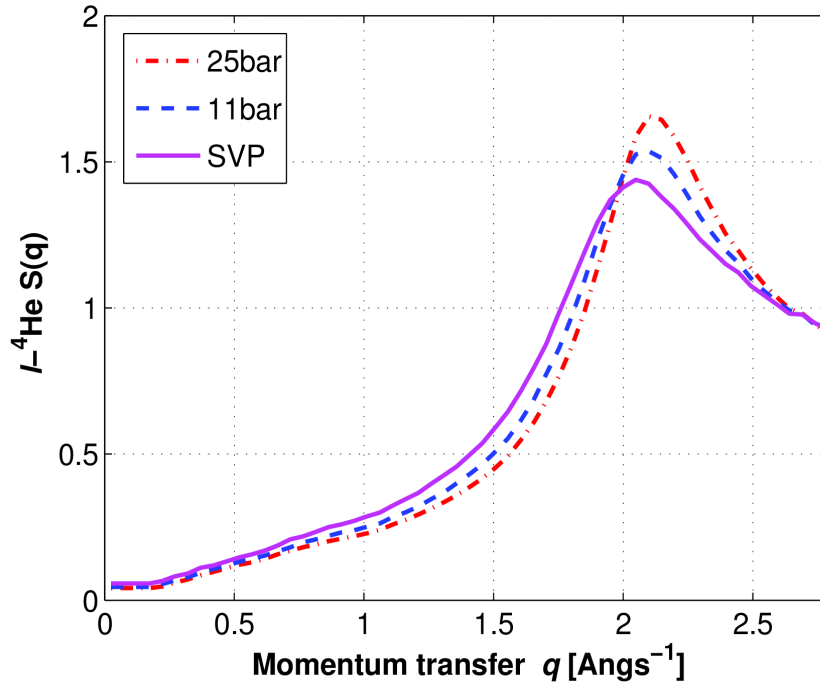


Illustration 2: $l\text{-}^4\text{He}$ structure factor $S(q)$ measured on IN5@ILL, with an incoming neutron wavelength $\lambda=3 \text{ \AA}$ [37].

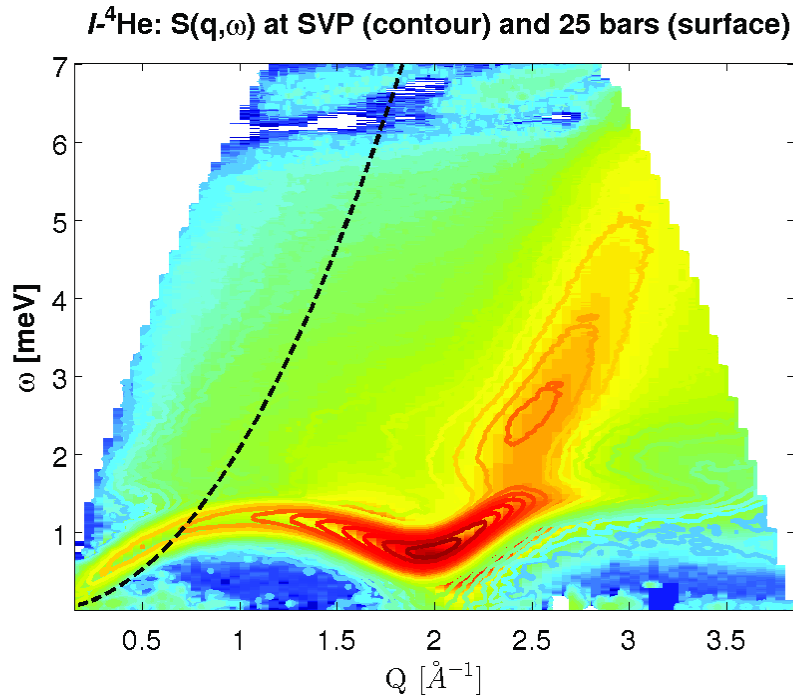


Illustration 3: Dynamic structure factor of liquid helium 4 at 1.7K as measured on IN5 at the ILL [37]. The contour lines indicate the SVP data, whereas the underlying surface displays the high pressure 25 bar data. Colours indicate $\log_{10}$ intensity levels from blue (low) to red (high). The free neutron dispersion is indicated as a black dashed line.

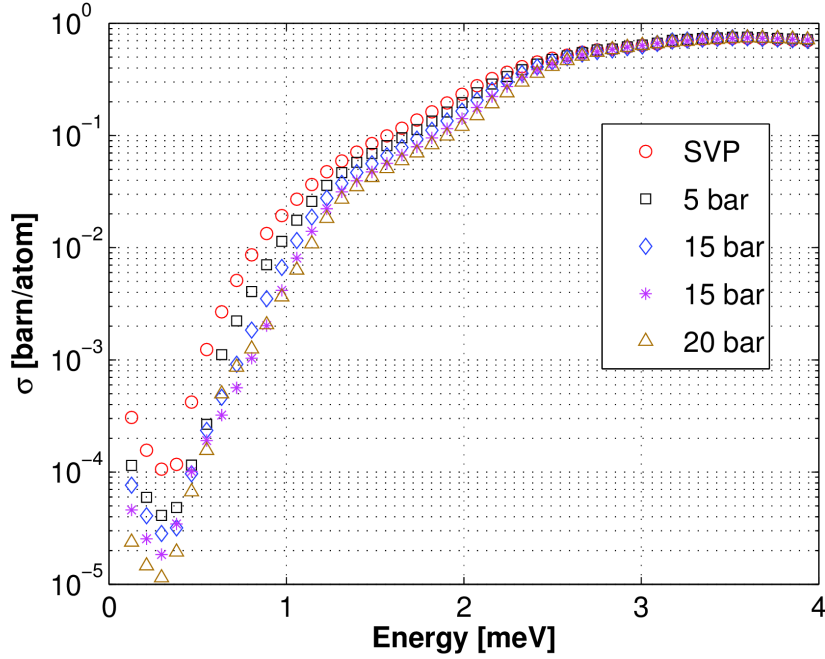


Illustration 4: Total scattering cross section of l - ^4He computed as a function of the incoming neutron energy, for p =SVP, 5, 10, 15 and 20 bar, obtained from [39].

3.1.2 The simulation model

We have assembled a l - ^4He moderator simulation using the McStas neutron scattering ray-tracing software. The model, as shown in Illustration 5, generates a spatially uniform neutron beam exiting from the H113 guide at the ILL. The velocity selector is accounted for by applying a triangular neutron wavelength distribution around its nominal value λ . The full wavelength spread was set to 14% below 6.5 Å and 8% above. The beam section is reduced to ϕ 30 mm after the velocity selector by mean of a perfect slit model and a flux monitor is located just in front of the He-II UCN target. The target is modelled as a cylinder with symmetry axis along the incoming beam, internal section ϕ 66 mm and length 20 cm. In this model, no aluminium housing nor reflecting coating is included, and the vessel is standing in the void. The target is surrounded by 4π solid angle detectors which are sensitive to the total scattered neutron events, the neutron events below 250 neV (corresponding to a beryllium coating inside the cell) on the He-II elementary excitation and on the multi-phonon continuum, and the multiply scattered neutrons below 250 neV. During simulations, the He pressure and incoming neutron wavelength are varied. The $S(q,w)$ data used by the *Isotropic_Sqw* component is that presented in the previous section, and takes into account the density variation with pressure. The cell temperature is set to $T=1$ K.

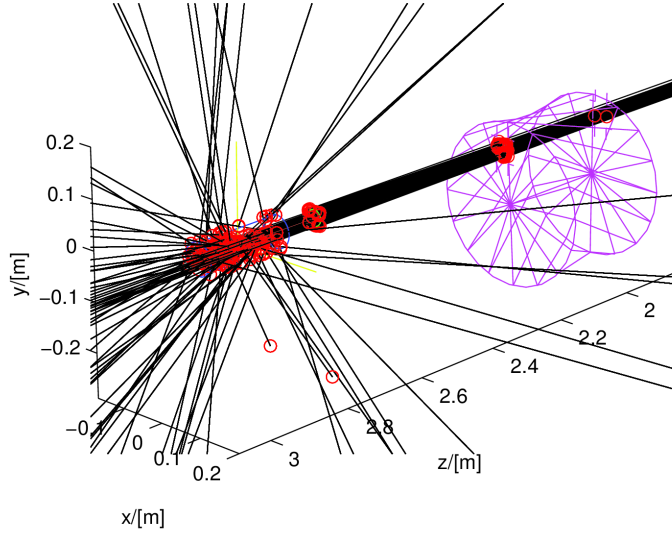


Illustration 5: Geometric representation of the ^4He UCN moderator model built with McStas. The neutron beam enters the velocity selector (on the right side), then passes a circular slit, and enters the liquid ^4He vessel (on the left side). Neutron trajectories are shown as black lines, whereas red circles indicate intersections with model components and scattering locations. The $\lambda=8.8 \text{ \AA}$ incoming neutrons are mostly scattered forward, and exhibit few multiple scattering events.

3.1.3 Simulation results

Using $l\text{-}^4\text{He}$ SVP data [39] at $T=1 \text{ K}$, we have simulated the total scattering around the moderator vessel, for $\lambda=8.8 \text{ \AA}$ incoming neutrons ($E_i=1 \text{ meV}$). The distribution of scattered neutron events is shown in Illustration 6 as a function of the scattering angle θ and final neutron energy E_f . The total fraction of scattered neutrons amounts to 0.9% of the incoming $\lambda=8.8 \text{ \AA}$ beam. Most neutrons are scattered below 20° angle and a final energy 0.7-0.8 meV. Other scattered neutrons gradually lose their energy while their average scattering angle increases. Neutrons with final energies below $40 \text{ \mu eV}$ are scattered mostly at angles between 50° and 70° , accounting for 0.062% of the incoming neutron flux. For $\lambda=5.7 \text{ \AA}$ incoming neutrons ($E_i=2.5 \text{ meV}$), the total scattered intensity is maximal, reaching 8.3 % of the incoming flux, but the proportion of UCN below $40 \text{ \mu eV}$ is then only of 0.008% of the incoming beam. With such low scattering efficiency, the beam attenuation (self shielding) is negligible in all cases.

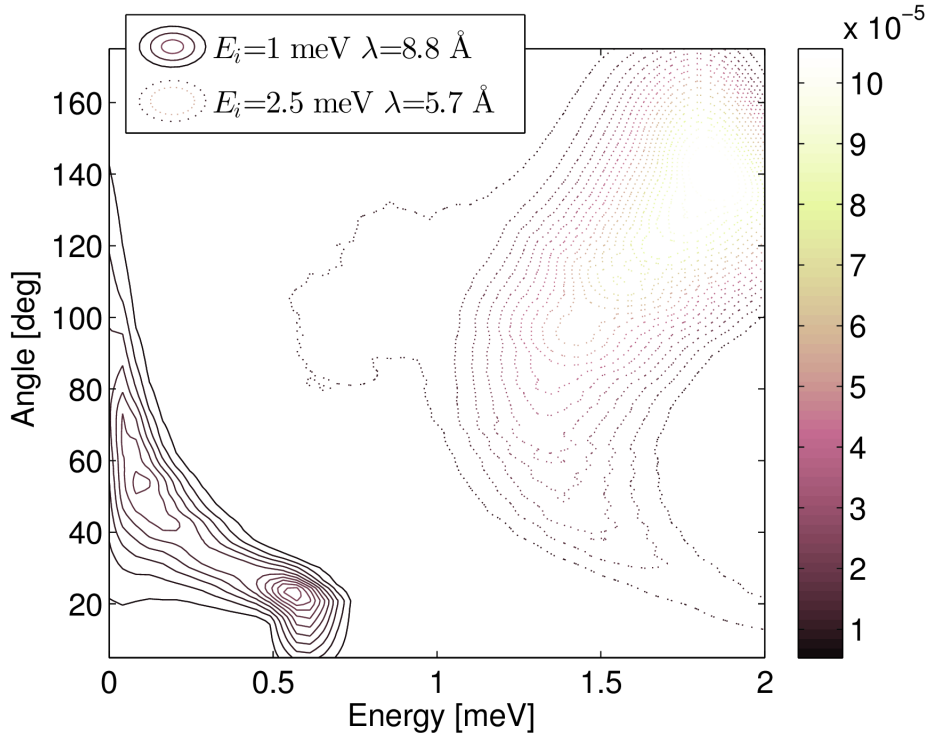


Illustration 6: Simulated distribution of scattered neutrons as a function of scattering angle θ and final energy E_f , for an incoming beam at $\lambda=8.8$ Å (continuous line) and 5.7 Å (dotted line) in a l - ^4He vessel at SVP. The colour scale on the right indicates the scattered intensity per 0.0465 [deg.meV] pixel normalised to the incoming flux.

We monitor the scattered neutrons with energies lower than 250 neV as a function of the liquid pressure and the neutron incident wavelength. In these events, we select those which correspond to $0.6 < q < 0.7$ Å and $\omega < 1.38$ meV, that is around the intersection of the elementary excitation with the free neutron dispersion (see Illustration 3). These events are labelled 'single phonon'. Other moderation processes involving the continuum are labelled as 'multi phonon'. All scattered events resulting from iterative scattering events in the cell are labelled as 'multiple scattering'.

The UCN production rate for incoming neutrons with wavelength $\lambda < 4.5$ Å is obtained using the IN5 data measured with $\lambda = 3$ Å [37]. The rest of the UCN experiment measurement range is obtained using IN6 data measured with $\lambda = 3$ Å [39]. For this data set, the SVP data was scaled up by 18% in intensity to account for the temperature change and the uncertainty on the maximum structure peak value $S(q)$, as suggested by [41][42]. Without this rescaling, the SVP data is clearly underestimated compared to all higher pressure results. As the experiments [37][39] had an energy resolution larger than that of the UCN production single phonon peak ($\lambda=8$ -10 Å), we used higher energy resolution data measured on IN5 with $\lambda = 5$ Å [38] to simulate the single phonon peak shape. These simulated data sets, on different wavelength ranges, are merged with iFit.

The simulated UCN production rate is shown in Illustration 7, and a closer view on the range $\lambda < 7.5$ Å, where most UCN are produced through multi-phonon processes, is shown in Illustration 8.

The sharp peak for $\lambda > 7.5$ Å corresponds to ultra-cold neutron produced by transferring the whole neutron energy (1 meV) to the elementary excitation in the liquid helium (single-phonon process).

We then extract the integrated intensity for $\lambda > 7.5$ Å as a function of the liquid pressure and compare it to its nominal value at $p=\text{SVP}$. The same comparison is achieved for the $\lambda < 7.5$ Å 'multi-phonon' range. These results are presented in Table 2 together with the first moment of the 'single phonon' peak λ^* . The multiple scattering contribution remains small at all pressures, around 5% of the multi-phonon process intensity and is only present for $\lambda < 6.5$ Å, as shown in Illustration 8.

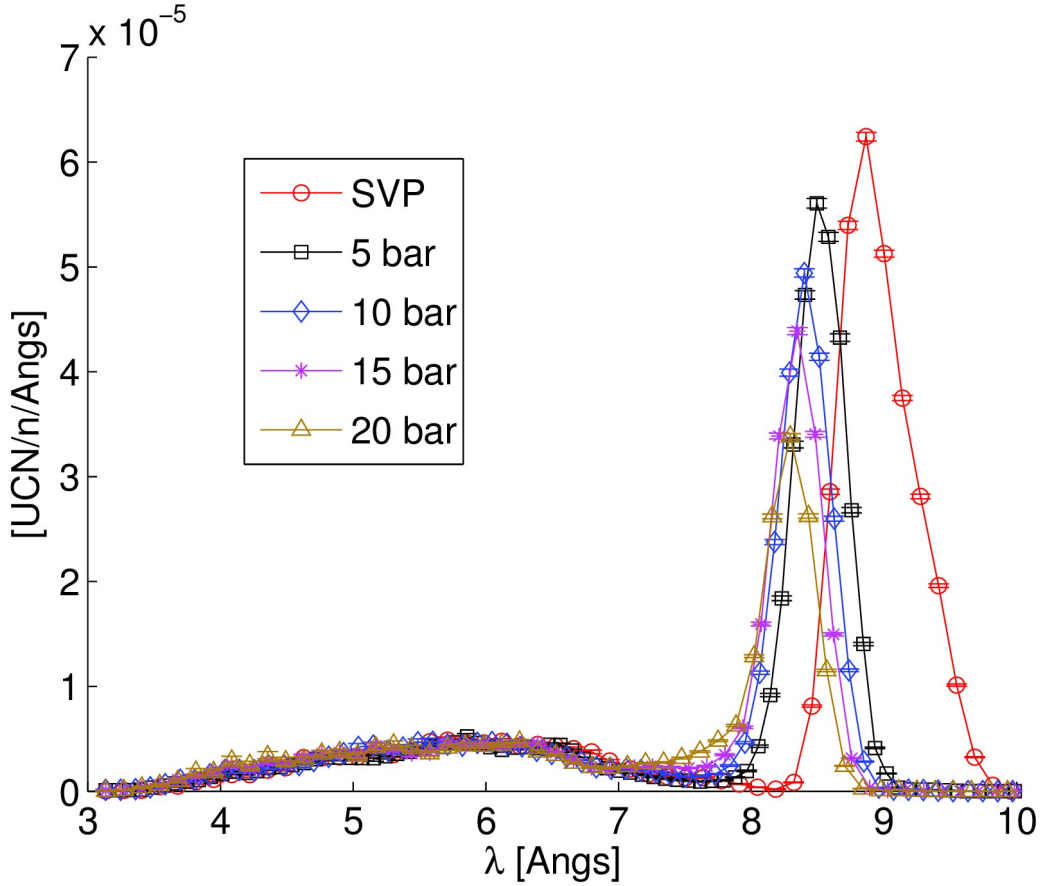


Illustration 7: Simulated wavelength resolved UCN production rate for $p=\text{SVP}$, 5, 10, 15 and 20 bar. The $l\text{-}^4\text{He}$ cell illuminated volume is 188 cm^3 . The lines are guides for the eye. There is no data correction nor fit in these results.

The multi-phonon estimated fraction depends on the choice for the cut-off wavelength, used as $\lambda_c = 7.5$ Å in Table 2. The Table 3 presents the integrated UCN production fractions from single-phonon, multi-phonon, and multiple-scattering processes. As seen in Illustration 7, the gap between the single-phonon and multiple-phonon UCN fractions vanishes when the cell pressure increases. Changing the cut-off value to $\lambda_c = 7.0$ Å to separate the integrals lowers the multi-phonon fraction by 1-3%, but does not significantly affect other quantities. The simulation results compare well with the previous experimental results [41], except for the multi-phonon fraction which is found constant

(close to 1) as a function of the cell pressure. Also, compared to this data, our simulated data set extends to the incoming thermal wavelength range $\lambda < 4.5 \text{ \AA}$, and we can label all neutron scattering events, and thus separate single and multi-phonon contributions more accurately.

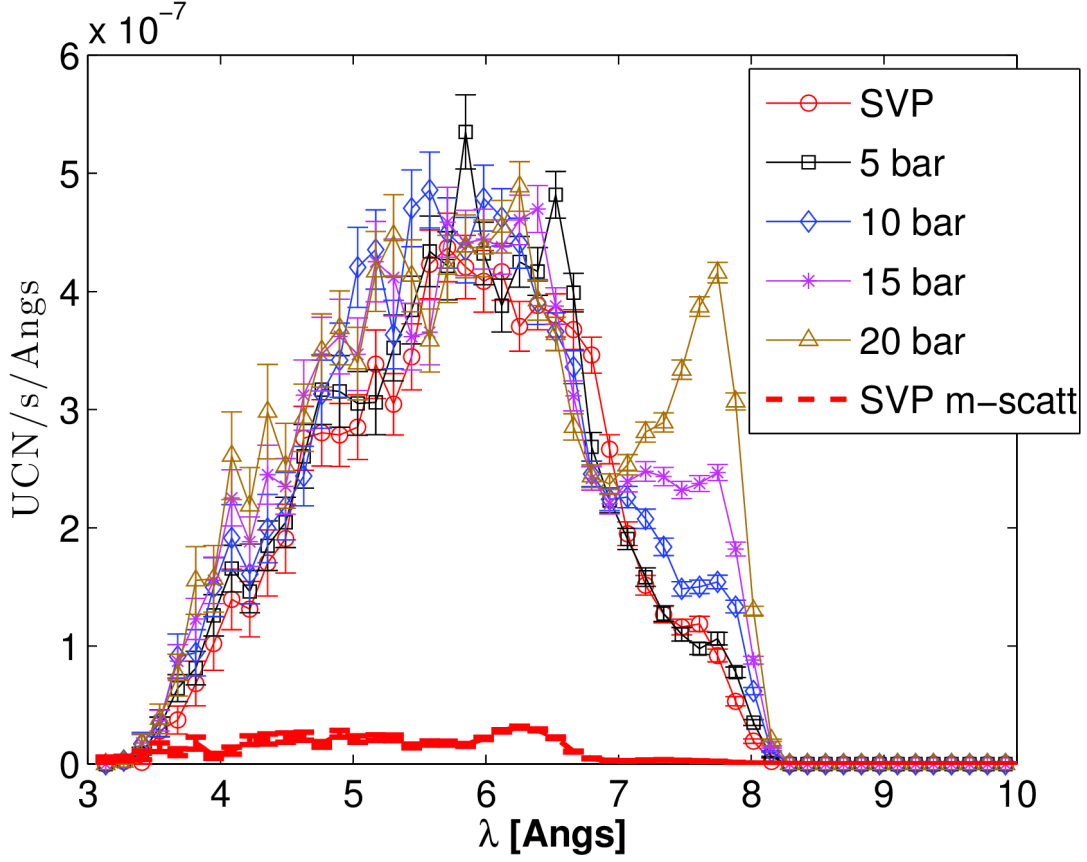


Illustration 8: Simulated UCN production per incident CN from multi-phonon processes for $p = \text{SVP}, 5, 10, 15$ and 20 bar . The multiple scattering contribution at $p = \text{SVP}$ is indicated as an underlying strong red line. The cell illuminated volume is 188 cm^3 . The lines are guides for the eye. There is no data correction nor fit in these results. The gap between single and multiple-phonon contributions is seen reduced above $\lambda = 7 \text{ \AA}$ with increasing pressure.

Absolute pressure [bar]	$s\text{-ph } \lambda^* [\text{\AA}]$	$s\text{-ph}/s\text{-ph(svp)}$	$m\text{-ph}/m\text{-ph(svp)}$	$m\text{scat}/m\text{-ph}$	$s\text{-ph } \lambda^* [\text{\AA}]$ (exp) [41]	$s\text{-ph}/s\text{-ph(svp)}$ (exp) [41]	$m\text{-ph}/m\text{-ph(svp)}$ (exp) [41]
SVP	8.96(1)	1	1	0.061(5)	8.92(2)	1	1
5	8.51(1)	0.66(1)	0.95(1)	0.056(4)	8.54(4)	0.68(4)	1.07(3)
10	8.40(1)	0.57(1)	1.01(1)	0.050(5)	8.39(2)	0.54(6)	1.16(3)
15	8.34(1)	0.50(1)	1.03(1)	0.051(5)	8.30(2)	0.46(5)	1.19(6)
20	8.29(1)	0.38(1)	1.06(3)	0.049(6)	8.26(2)	0.40(3)	1.22(6)

Table 2: Simulated single-phonon peak position λ^ , and UCN production rates for various pressures relative to the SVP production rate, from single-phonon (s-ph), and multi-phonon (m-ph)*

processes, as well as the multiple-scattering (mscat) proportion in the multi-phonon production. Experimental data from [41] are indicated in the last three columns.

<i>Absolute pressure</i>	<i>s-ph ($\lambda > 7.5 \text{ \AA}$)</i>	<i>m-ph ($\lambda < 7.5 \text{ \AA}$)</i>	<i>multi-scat. ($\lambda < 7.5 \text{ \AA}$)</i>
<i>[bar]</i>	<i>UCN/n/s 10^{-7}</i>	<i>UCN/n/s 10^{-7}</i>	<i>UCN/n/s 10^{-8}</i>
SVP	5.94	1.60	1.05
5	3.95	1.53	0.91
10	3.36	1.63	0.86
15	2.95	1.68	0.90
20	2.27	1.72	0.89

Table 3: Simulated UCN production absolute integrated intensities for various pressures, for the H113 beam line at the ILL and a 188 cm^3 $l\text{-}^4\text{He}$ cell.

From Illustration 7 and Illustration 8, we notice that for high pressure, no clear separation can be made between single and multiple phonon moderation processes, as in fact the elementary excitation (maxon) approaches the multi-phonon continuum while the sound velocity increases, shifting the intersection point of the phonon line with the free neutron dispersion towards higher energies. Simultaneously, the scattering intensity on $S(q, \omega)$ significantly drops, resulting in a lower UCN production rate.

It appears that the UCN production rate is not improved by increasing the liquid pressure and density. The total scattering cross section for high pressures is lower than that for $p=\text{SVP}$. The best UCN production rate is achieved with about $3.2 \cdot 10^{-8}$ UCN/n/ $\text{\AA}/\text{cm}^3$ per $\lambda=8.9 \text{ \AA}$ incident neutrons (see Illustration 7). In practice, the UCN production rate is highly variable around the single-phonon peak, so that the effective integrated UCN production rate over the monochromatising device wavelength band for incoming neutrons is smaller than its maximum by a factor e.g. 2 (for a triangular transmission). We then use as a mean single-phonon UCN production rate $1.6 \cdot 10^{-8}$ UCN/n/ $\text{\AA}/\text{cm}^3$ for $8.9 \text{ \AA}$ incident neutrons.

3.1.4 UCN production rate estimates for ILL experiments

We now estimate the UCN production rate on a variety of experimental locations at the ILL, using such a moderator. These estimates can be considered as an upper limit, as additional losses in the cell UCN extraction system and imperfections of the neutron optics are here neglected.

The ^4He cell installed at the end of the current H53 guide at the ILL providing a flux of $2.6 \cdot 10^7$ n/s/ $\text{cm}^2/\text{\AA}$ and monochromated with a $\delta\lambda/\lambda=8\%$ velocity selector around $\lambda=8.9 \pm 0.35 \text{ \AA}$, we get a UCN production rate of 1.9 UCN/s/ cm^3 . An experiment by Baker *et al* [43] has reported a production rate around 0.9 UCN/s/ cm^3 in similar conditions. The difference probably arises from our model, which considers a perfect cell coating, detector efficiency, velocity selector, and flux distribution.

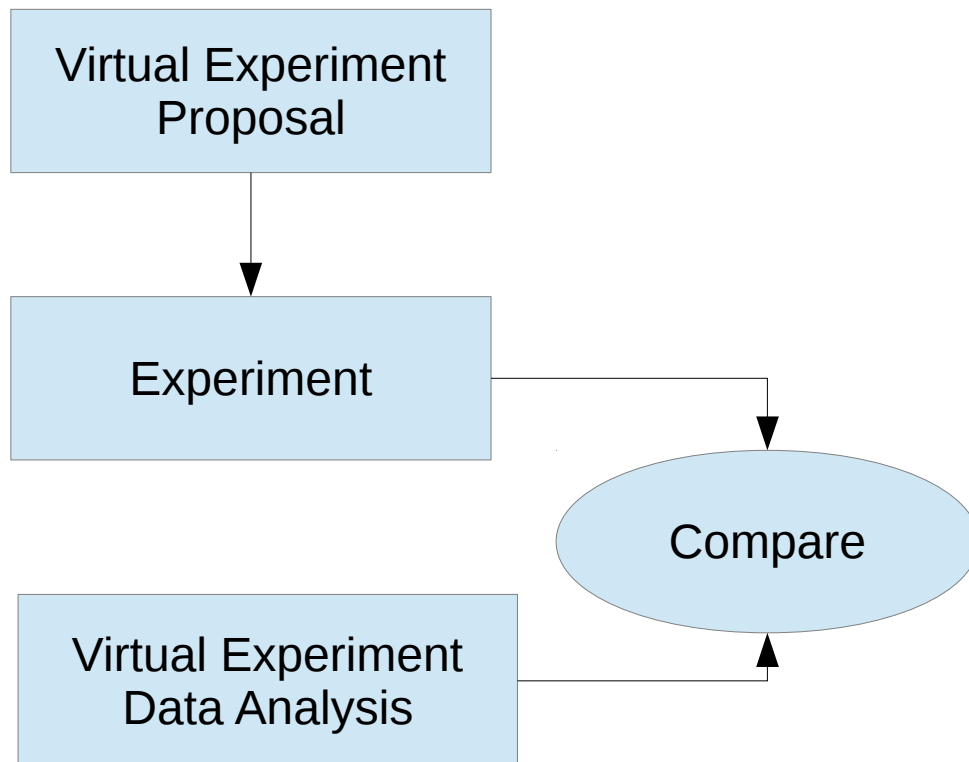
Installing an intercalated pyrolytic graphite monochromator with reflectivity 50% at the H172 cold beam at the ILL for the SuperSUN and GRANIT experiments, we get an incident flux of about $7 \cdot 10^7$ n/s/cm² around $\lambda=8.9 \pm 0.19$ Å which generates 4 UCN/s/cm³.

Using the PF1b experimental hutch with an incoming flux estimated to $2.4 \cdot 10^8$ n/s/cm² (ILL_H113 McStas model provided as an example with the software) from a $\delta\lambda/\lambda=8\%$ velocity selector around $\lambda=8.9 \pm 0.35$ Å, we get a UCN production rate of 25 UCN/s/cm³.

Now turning to the CryoEDM experiment to be installed on the H523 guide at the ILL, we estimate from a full H5 guide simulation that the un-polarised beam entering the experiment will be about $2 \cdot 10^9$ n/s/cm²/Å from a $\delta\lambda/\lambda=8\%$ velocity selector around $\lambda=8.9 \pm 0.35$ Å. We then obtain a UCN production rate of 193 UCN/s/cm³ in the ⁴He cell. When adding the CryoEDM polariser, we obtain a UCN production rate of 29 UCN/s/cm³ for an incoming polarised beam flux of $2.8 \cdot 10^8$ n/s/cm²/Å at 8.9 Å.

3.2 Experiment planning: the dynamics of liquid indium studied with a triple axis spectrometer

In this section, we demonstrate how a virtual experiment can be used to support an experiment proposal, as well as during the acquisition, and for the final data analysis. The schematic representation of the methodology presented below is shown as Drawing 2.



Drawing 2: Using virtual experiments to motivate a proposal, help during the experiment, and analyse the data.

3.2.1 The liquid Indium molecular dynamics simulation

The Vienna Ab-initio Simulation Package (VASP) [18] is a density functional theory (DFT) code [44] using the local density approximation (LDA) [45]. The projector augmented wave potentials (PAW) [46] with the Ceperley-Alder exchange correlation functional (CA) [47] have been employed to describe electronic states. For indium, the pseudo-potential energy cut-off is 96 eV. The software in version 5.2.8 has been used in its molecular dynamics mode, which determines iteratively the atom positions, and a Brillouin zone sampling around the Gamma point only (for liquids).

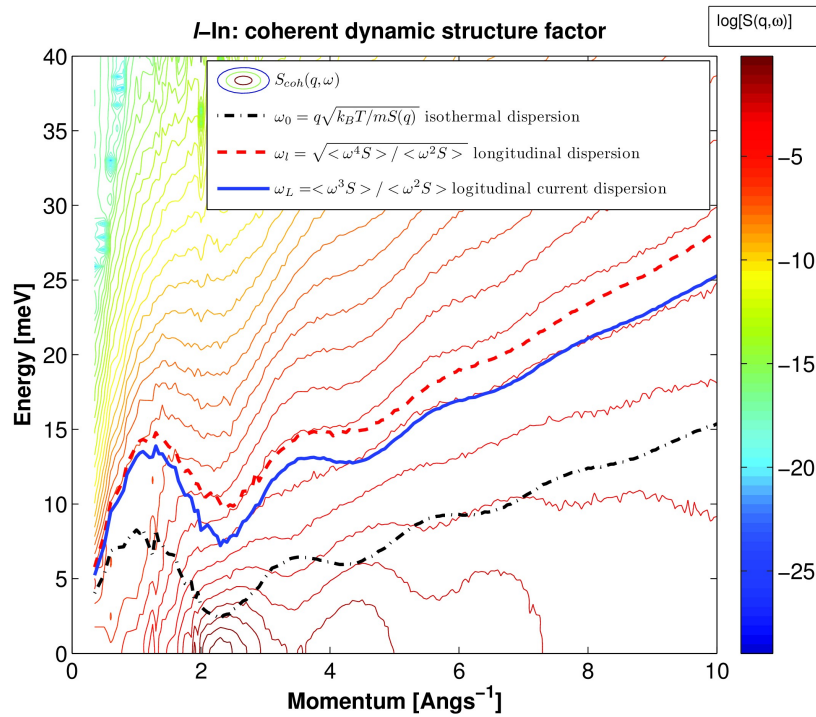


Illustration 9: The coherent classical dynamic structure factor $S(q, \omega)$ of liquid indium at $T=741K$ obtained from ab-initio molecular dynamics simulation. The isothermal ω_0 , longitudinal ω_l and apparent ω_L sound velocities computed from the $S(q, \omega)$ are also shown. The notation $\langle \omega^n S \rangle$ is the n -th moment of the scattering law.

The liquid indium molecular dynamics simulation has been performed using 200 indium atoms in a 17.71 Å edge length cube, corresponding to a density $\rho=6.86$ g/cm³ and an atom density of $\rho_0=0.0359$ at/Å³. This configuration defines the smallest momentum exchange between neutrons and gold to $q=2\pi/17.71=0.35$ Å⁻¹, and the intrinsic energy resolution to 0.05 meV. The molecular dynamics simulation was first equilibrated in the (N,V,T) canonical ensemble around $T=750$ K, for 9 ps (3000 steps of 3 fs). The thermalised atom distribution was then simulated in the (N,V,E) microcanonical ensemble for a duration up to 36 ps, as 12000 time steps of 3 fs. The temperature during the trajectory remained in the range $T=741\pm 29$ K. Simulation carried out at lower temperature and with other pseudo-potentials (PAW GGA PBE) resulted in the formation of crystal precursors in the liquid, with strong Bragg peaks in the structure factor $S(q)$, as already reported by Kresse [19]. The diffusion coefficient for this temperature, computed from the mean square

displacement vs. time, was estimated from the trajectories as $D=4.75(4) \cdot 10^{-5} \text{ cm}^2/\text{s}$. This value is slightly under estimated when compared to previous data [48].

The dynamic structure factor $S(q, \omega)$ of indium was obtained from the trajectories by Fourier transform in space and time, using the nMoldyn/MMTK software [23], both for the coherent and incoherent processes, in the so-called classical approximation. The coherent contribution is shown in Illustration 9. The momentum and energy smoothing filters used in the fast correlation algorithm were set to $0.05 \text{ \AA}^{-1}$ and 1 meV respectively. These quantities define an effective resolution assigned to the dynamic structure factor. The accuracy on the resulting $S(q, \omega)$ values for 36 ps simulation with 12000 iterations is within 10%. The scattering law is computed up to $q=10 \text{ \AA}^{-1}$ and $\omega=\pm 150 \text{ meV}$ momentum and energy transfer.

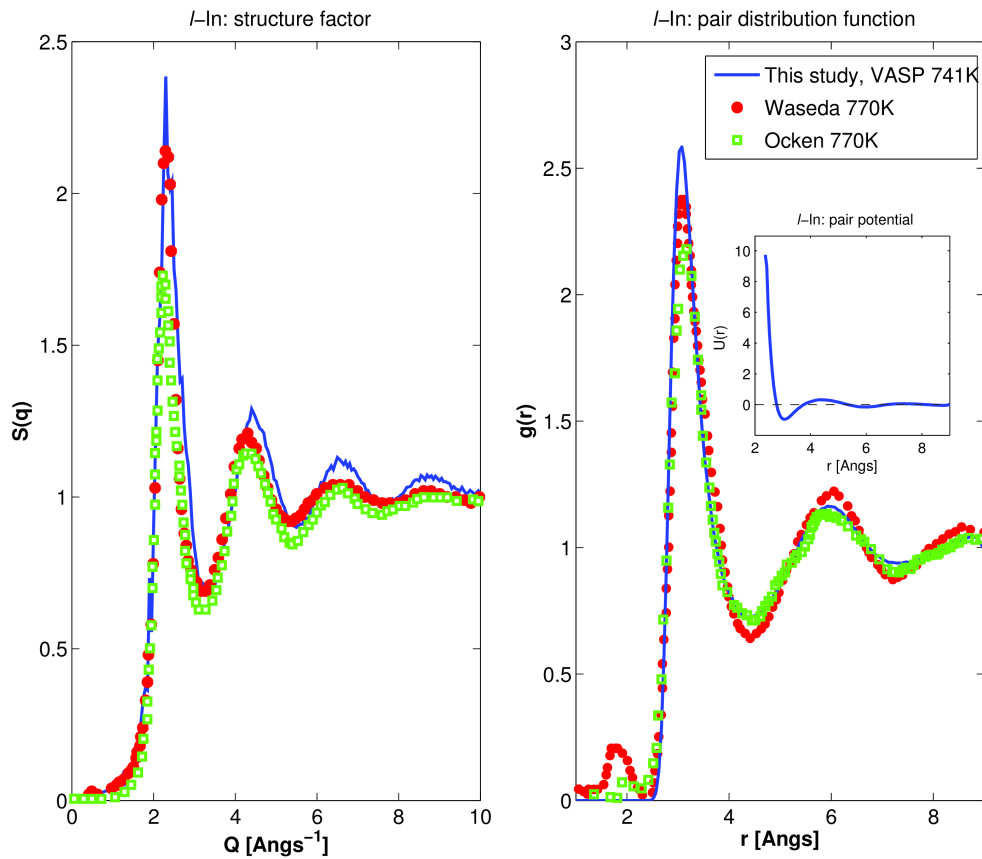


Illustration 10: The static structure factor $S(q)$ (left) and pair distribution function $g(r)$ (right) of liquid indium at $T=741\text{K}$ obtained from ab-initio molecular dynamics simulation, compared to experimental data [49][50]. The right inset shows the pair potential $U(r)$ deduced from the pair distribution function.

The integration of the coherent contribution over the energy axis provides the static structure factor

$$S(q) = \int_{-\infty}^{\infty} S(q, \omega) d\omega ,$$

which is shown in Illustration 10. The maximum of the first sharp structure peak is found at $q_p=2.38 \pm 0.02 \text{ \AA}^{-1}$, and subsequent peaks are found around 4.5 and $6.6 \pm 0.05 \text{ \AA}^{-1}$. The comparison with

previous X-ray diffraction results at 770 K [49][50] reveals differences in the shoulder of the first sharp peak, as well as in the position and amplitude of the further oscillations. However, we notice that there are significant differences even within experimental results. Also, as we shall mainly study the structure and dynamics around the first sharp peak, we shall consider in the following that the simulation result is satisfactory. As stated previously, a stable and realistic molecular dynamics simulation of liquid indium is particularly difficult to obtain.

From the static coherent structure factor, we can extract the pair distribution function (PDF)

$$g(r) = 1 + \frac{1}{2\pi^2 r \rho_0} \int_0^\infty q [S(q) - 1] \sin(qr) dq ,$$

which is shown in Illustration 10 and compared to experimental data [49][50]. The PDF is directly related to the pair potential $U(r)/k_B T = -\ln[g(r)]$ shown as inset in the figure. The mean distance to the first neighbour is found at $r = 3.05 \pm 0.02$ Å, and the first minimum is found at $r_{min} = 4.37 \pm 0.02$ Å. The corresponding coordination number determined from the integration of the radial distribution function (RDF), $4\pi^2 \rho_0 g(r)$, up to r_{min} is found as CN=11.8.

From the classical dynamic coherent structure factor, we can compute the collective dispersion and sound velocities [34]. In the following, we denote $\langle \omega^n S \rangle$ the n-th moment of the coherent scattering law as a function of the momentum

$$\langle \omega^n S \rangle = \int_{-\infty}^{\infty} \omega^n S(q, \omega) d\omega$$

As the classical scattering law S is symmetric in energy, the odd moments are null. The zero-th moment is the structure factor, $\langle S \rangle = S(q)$. The second moment of the $S_{coh}(q, \omega)$, which converges to the squared recoil energy, provides the isothermal dispersion $\omega_0(q)$ and sound velocity

$$\omega_0(q) = \sqrt{\frac{\langle \omega^2 S \rangle}{S(q)}} = q \sqrt{\frac{k_B T}{m S(q)}} \quad \text{and} \quad c_T = \frac{\omega_0(q)}{q} ,$$

where $m = 115$ g/mol is the indium atom mass. The ratio of the 4th to 2nd moment provides the longitudinal dispersion $\omega_l(q)$ and corresponding sound velocity

$$\omega_l(q) = \sqrt{\frac{\langle \omega^4 S \rangle}{\langle \omega^2 S \rangle}} \quad \text{and} \quad c_l(q) = \frac{\omega_l(q)}{q} .$$

However, computing this quantity is subject to strong uncertainties as the simulated dynamic structure factor is a noisy function. In order to estimate the apparent phonon dispersion, we compute the maxima $\omega_L(q)$ of the longitudinal current correlation function $J_L(q) = \omega^2 S(q, \omega)/q^2$ [51][52], namely (evaluated on the positive energy range only)

$$\omega_L(q) = \frac{\langle \omega^3 S \rangle}{\langle \omega^2 S \rangle} \quad \text{and} \quad c_L(q) = \frac{\omega_L(q)}{q} ,$$

which are close to $\omega_l(q)$ and $c_l(q)$. These three dispersions are reported as lines in Illustration 9. The longitudinal and apparent dispersions are above the isothermal one in energy. We then extract the corresponding sound velocities, as shown in Illustration 11 and find an extrapolated adiabatic value in the hydrodynamic regime $q \rightarrow 0$ as $c_s = 2289 \pm 27$ m/s in agreement with previous work [53][52][54][55]. The isothermal sound velocity is found as $c_T = 1697 \pm 23$ m/s.

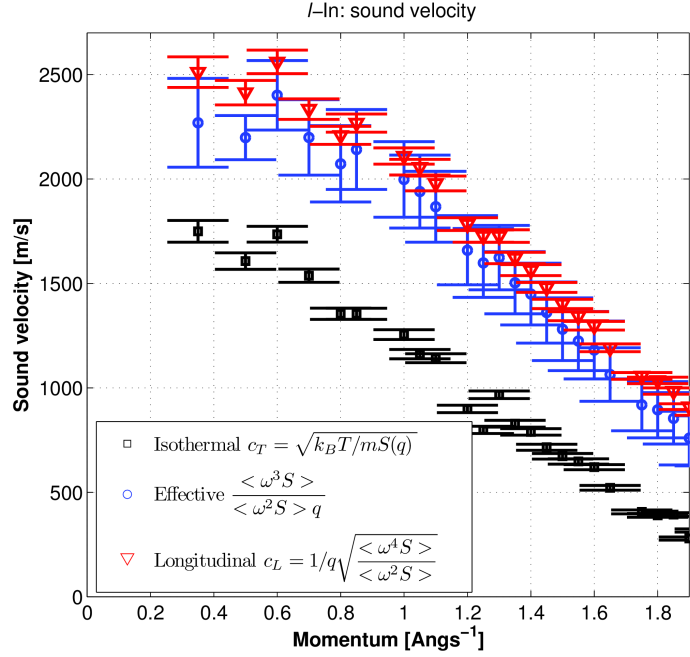


Illustration 11: The sound velocities extracted from the collective dispersions of liquid indium at $T=741\text{K}$ obtained from *ab-initio* molecular dynamics simulation. The notation $\langle \omega^n S \rangle$ is the n -th moment of the scattering law.

3.2.2 The structure and dynamics of *l*-In measured around the first sharp peak with the IN22 TAS

Based on a triple-axis spectrometer virtual experiment with the previous dynamic structure factor $S(q, \omega)$, an experiment proposal was submitted with estimates of the quasi-elastic contribution intensity achievable during the requested beam time. On acceptance, we have measured the dynamic structure factor of liquid indium at $T=620 \text{ K} \pm 50$ (indium melts at 429 K) in the momentum transfer q range 1.5 to $3.5 \text{ \AA}^{-1}$ and energy transfer ω range -10 to 10 meV .

The thermal triple axis spectrometer IN22 is installed at the end of the super-mirror coated guide H25 at the ILL. The instrument was used in un-polarised mode, with an incoming constant neutron wave-vector $k_i=3.84 \text{ \AA}^{-1}$ ($E_i=30.56 \text{ meV}$) and a graphite filter to suppress monochromator higher orders. The monochromator and analyser were focusing multi-slab pyrolytic graphite ensembles ($d=3.355 \text{ \AA}$). Slits were adjusted before and after the sample to reduce the gauge volume to the sample size and thus limit the contribution of the scattering from the sample environment.

As the scattering cross sections of indium are low ($\sigma_{\text{coh}}=2.08 \text{ barn}$, $\sigma_{\text{inc}}=0.54 \text{ barn}$) and absorption cross section is high ($\sigma_{\text{abs}}=193.5 \text{ barn}$ at $\lambda=1.78 \text{ \AA}$), we employed a gas-flow levitation furnace with laser heating as described in [56] to avoid the use of a sample container that would scatter much more than the indium sample itself. An argon gas flow was injected from the bottom through a conical B_4C nozzle with central gas channel to levitate the sample about 1 mm above the nozzle. The sample was melt with a single CO_2 laser hitting a 4 mm diameter indium ball from the top. The temperature was continuously monitored with a pyrometer. The total measurement time was two weeks.

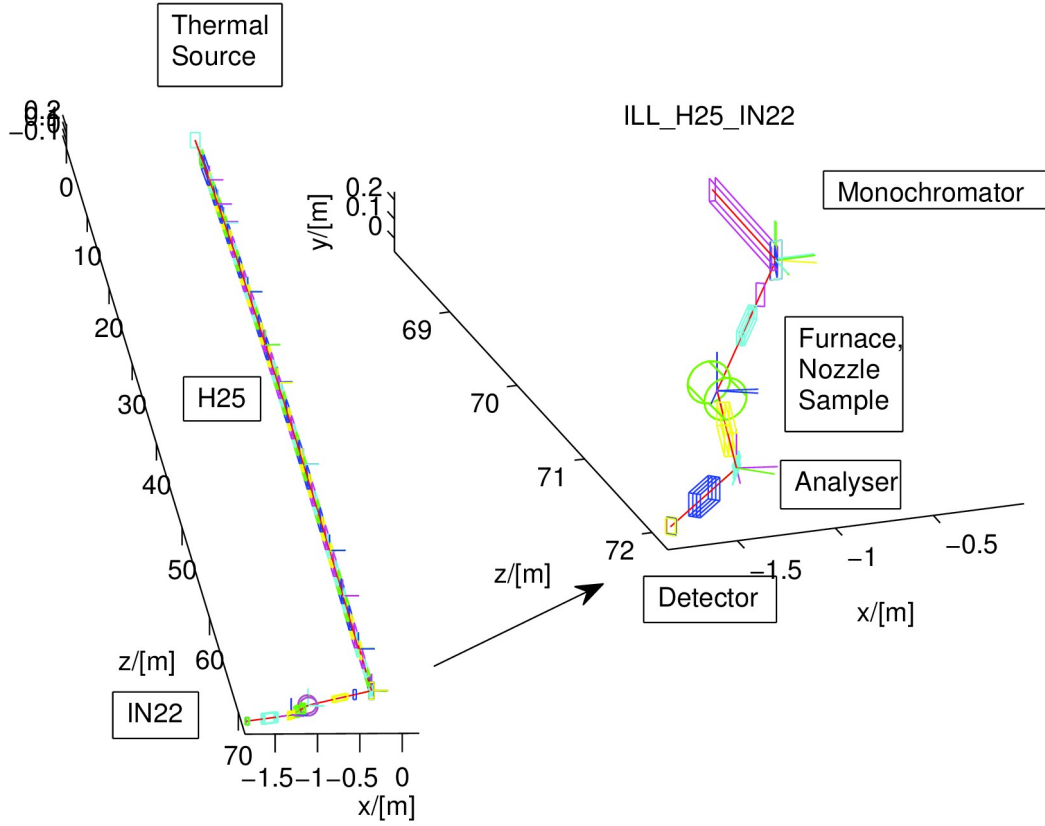


Illustration 12: The IN22 thermal triple-axis spectrometer at the ILL, modelled with McStas. The neutrons exiting the reactor enter the H25 guide (left), then reach the IN22 focusing monochromator position, the sample above the gas nozzle and furnace cylinder, the analyser, and finally the detector.

The corresponding instrument description for McStas is shown in Illustration 12. The model includes all parts from the instrument, the sample itself (liquid sphere with the *Isotropic_Sqw* component), and the sample environment (furnace chamber in aluminium, B₄C nozzle below the sample). This instrument description is part of the McStas distribution.

The levitation furnace ensures the lowest sample environment scattering contribution (no materials around except the B₄C nozzle), and a very low chemical reaction in the sample vessel. The nozzle and sample are positioned at the centre of an aluminium cylindrical vessel, pumped and filled with high purity argon gas. Indium does not react with argon, but is known to oxidise when exposed to air [57]. For a $k_i = 3.84 \text{ \AA}^{-1}$, the mean neutron penetration depth in indium is $1.78/\rho\lambda\sigma_{\text{abs}} = 1.5 \text{ mm}$. Thus, an oxidized layer, that may form from argon gas impurities, should only account for a small fraction of the scattered signal. To our knowledge, this is the first report of a neutron inelastic scattering experiment making use of a levitation furnace.

During the experiment, we noticed an additional scattering contribution to the expected liquid indium structure, which first sharp structural peak appeared broadened to higher momentum values. The virtual experiment, launched in parallel with the acquisition, suggested a measurable scattering

from the B₄C nozzle below the sample, which would diffract at $q=2.44$ and $2.63 \text{ \AA}^{-1}$. The background measurement confirmed a structured signal, which is made visible for the very weak indium scattering.

The experimental data set is corrected by

- the resolution volume $R_A(k_f) k_f^3 / \tan \theta_A$ with R_A being the analyser reflectivity, k_f the final neutron wave-vector and θ_A the analyser rotation angle ;
- the detector efficiency approximated as $1 - e^{-0.0591 P \lambda_f \phi}$ with the neutron wavelength λ_f in [$\text{\AA}$], the gas pressure P in [bar] and the tube diameter ϕ in [cm] ;
- the empty furnace (without the sample, but with the nozzle), which is subtracted ;
- the incoherent contribution estimated from the virtual experiment (as it was not measured).

The contribution of the furnace alone, without the B₄C nozzle, is not measurable as it falls outside of the measurement gauge volume.

Using the ResCal/ResLib code available in iFit [11][12] with the Popovici method, we estimate the momentum instrument resolution to $\delta Q=0.03 \text{ \AA}^{-1}$ (FWHM) for the IN22 configuration used during the acquisition (fixed $k_f=3.84 \text{ \AA}^{-1}$), as shown in the Illustration 13. The energy resolution is estimated to $\delta E=1.3 \text{ meV}$ on a phonon line (flat dispersion).

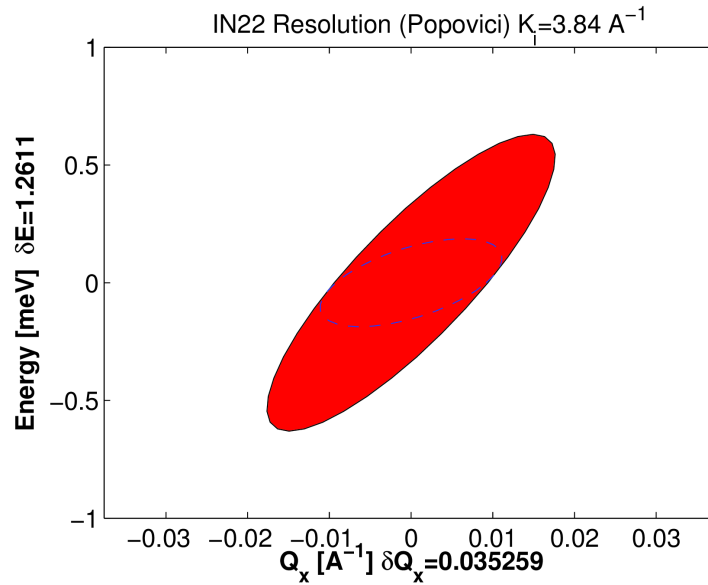


Illustration 13: Transverse momentum and energy resolution computed for the IN22 spectrometer in the configuration used during the experiment, computed with iFit/ResLibCal [11]. The dotted ellipse indicates the intersection of the ellipsoid with the (Q_x, E) plane.

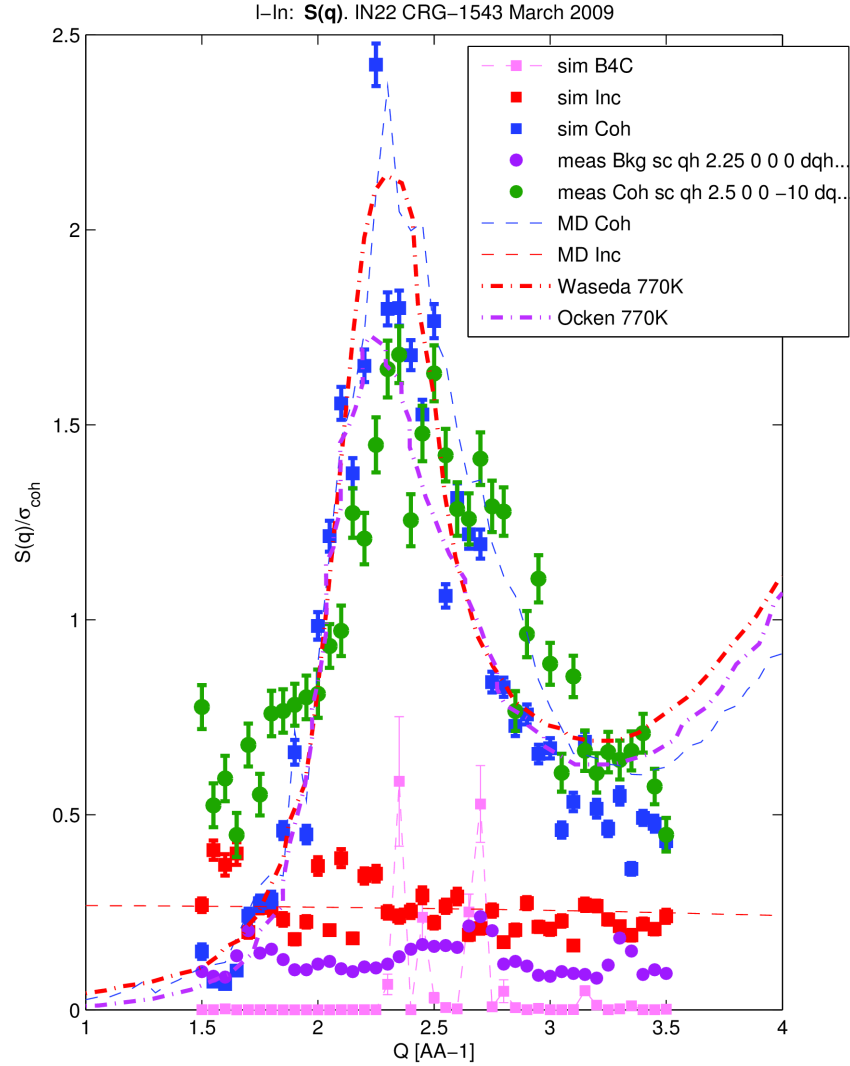


Illustration 14: Structure factor of liquid indium from the experiment ($T=620\text{K}$, circles), from the virtual experiment ($T=741\text{K}$, squares), from molecular dynamics ($T=741\text{K}$, dashed) and from previous studies [53][49] (dash-dot). The incoherent contribution (red) is scaled to the coherent one (blue) for direct intensity comparison.

The structure factor $S(q)$ computed on the dynamic range acquired during the experiment is shown in Illustration 14, and compared to previous data. There is a significant dispersion among the results, just as mentioned earlier in section 3.2.1. The acquired data suffers from the low statistics, resulting from the very low scattering and the high absorption cross sections. The simulated incoherent contribution agrees with the molecular dynamics $S_{inc}(q)$. We can measure the effect of the virtual experiment by comparing the simulated coherent structure factor (blue squares) with the one extracted from the molecular dynamics modelling (blue dashed line). The global shape of the structure peak is retained, within the error bars. Compared with previous studies [49][53], the coherent part of the measurement, obtained after removal of the incoherent and background

contributions from the total measured signal, exhibits a tail on the larger momentum side, by about $0.1 \text{ \AA}^{-1}$. This tail can hardly be associated with a resolution or misalignment of the spectrometer (see Illustration 13). It may be related to temperature gradients inside the sample, where the heating is sent from the top laser, but the gas flow below the sample locally cools down the lower part of the sample. Indeed, crystallised indium precursors would then diffract at $q=2.31, 2.54$ and $2.73 \text{ \AA}^{-1}$. The measured structure peak is also in reasonable agreement with the molecular dynamics result.

We point out that integrated quantities, such as $S(q)$, should in principle extend on a large energy range around the elastic line, as detailed in sections 2.1 and 3.2.1. In practice, as the acquisition rate for liquid indium was low due to its very small scattering cross section and large absorption, the measured energy range was limited. Consecutively, the structure factor obtained from the integration on the measured dynamic range should be smaller than the one obtained e.g. from X-rays [49][50]. This approximation due to a limited integration range also stands for other integrated intensities, such as the scattering law moments $\langle \omega^n S \rangle$ used to compute the collective dispersions. The Illustration 14 also indicates a broader simulated structural peak compared to X-ray measurements, but this is a consequence of the molecular dynamics (as mentioned in section 3.2.1 and Illustration 10).

We now turn to the inelastic scattering from liquid indium. The Illustration 15 presents the whole measurements (the upper three surfaces), as well as the corresponding simulated data from the virtual experiment. The measured coherent contribution (upper surface) is obtained by subtracting the simulated incoherent and the background measurement from the raw measurement.

In order to evaluate the quality of these data sets, we have extracted the collective dispersion quantities, as detailed in section 3.2.1. The moments $\langle \omega^n S \rangle$ of the scattering law are computed on the measured dynamic range, as a function of the momentum exchange. The isothermal dispersion $\omega_0(q) = q \sqrt{k_B T / m S(q)}$ can be obtained directly from the inverted structure factor, as the second moment is the squared recoil energy [34]. However, the integration range is limited by the actual measurement and the effective second moment is smaller than its nominal value. The resulting effective isothermal dispersion $\omega_c(q) = \langle \omega^2 S \rangle_{-10\text{meV}}^{+10\text{meV}}$ is shown in Illustration 16. Close to the first sharp structural peak (around $q=2.2 \text{ \AA}^{-1}$), the effective isothermal frequency gets closer to the nominal value, due to the De Gennes narrowing which limits the restricted integration range effect (the signal is gathered around the first sharp peak and the weight of the non-measured integral [10 meV,] is reduced). In addition, the longitudinal current frequency is also shown. The simulated data set appears in good agreement with the theoretical value obtained from the molecular dynamics. This demonstrates that the virtual experiment does not degrade substantially the structural and dynamical information from the sample. However, except for the estimate of the nominal isothermal frequency $\omega_0(q)$ which only depends on $S(q)$, the other measured dispersion frequencies are larger than the expected ones, probably because the measured structure factor is itself underestimated (see Illustration 14) and the integration range should be larger.

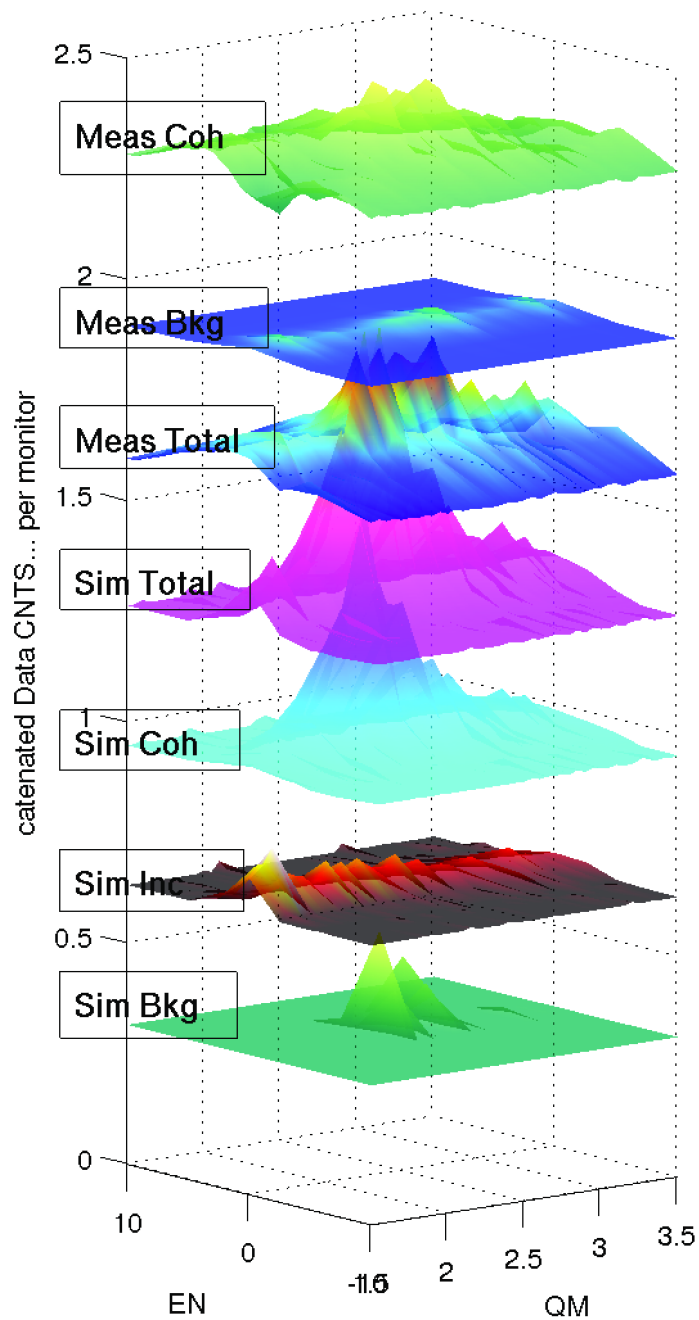


Illustration 15: The measured signal acquired on the IN22 spectrometer, compared with the simulated virtual experiment. The data set is decomposed into the coherent, incoherent, and background signals.

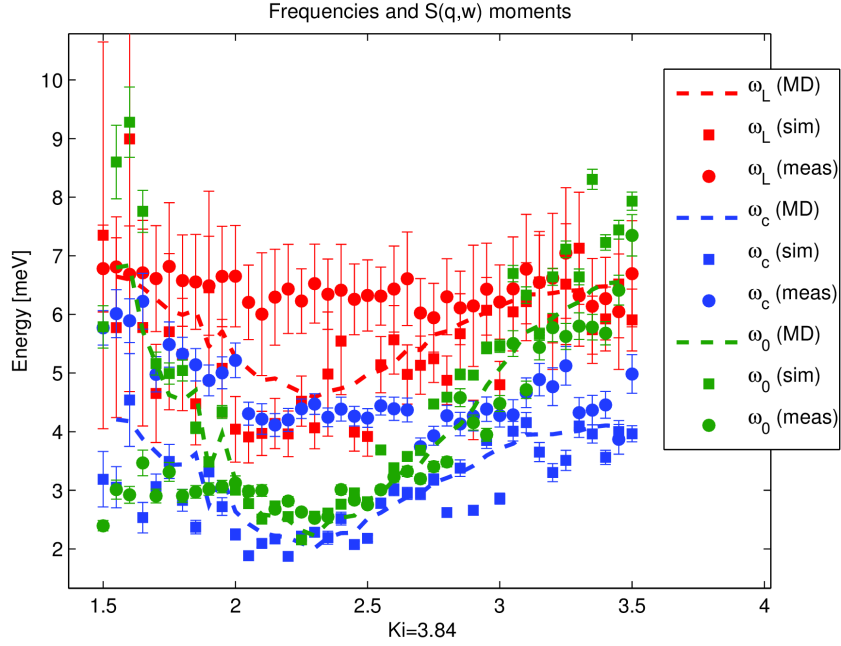


Illustration 16: The collective dispersions in liquid indium, as measured (circles) on the IN22 spectrometer and computed from ab-initio molecular dynamics (dashed lines), and simulated from the virtual experiment (squares). $\omega_L = \langle \omega^3 S \rangle / \langle \omega^2 S \rangle$ is the longitudinal current dispersion (red). The isothermal dispersion is computed from both the normalised second frequency moment $\omega_0 = \sqrt{\langle \omega^2 S \rangle / S(q)}$ (in blue) and from $\omega_c = q \sqrt{k_B T / m S(q)}$ (in green).

For this study, we have managed to perform a molecular dynamics simulation of liquid indium, which models are known to crystallise easily [18]. We have demonstrated that the virtual experiment does not affect significantly the sample structural and dynamical information from the molecular dynamics, and this is certainly the case as well for real experiments (hopefully). The proposed experiment was carried out, but as expected the measurement statistics is very low, which explains why it was never measured with neutron spectrometers before. In addition, we have used the levitation furnace for inelastic scattering for the first time on a neutron spectrometer, and demonstrated that a sensible signal can be obtained. The measured first structure peak coincides with the previous studies, as well as the theoretical $S(q)$ from molecular dynamics. It is possible to extract the collective dispersions from the inelastic data measurements, and the virtual experiment indicates that these dispersions are indeed measurable.

3.3 Data analysis with embedded virtual experiment: application to powder diffraction Rietveld refinement

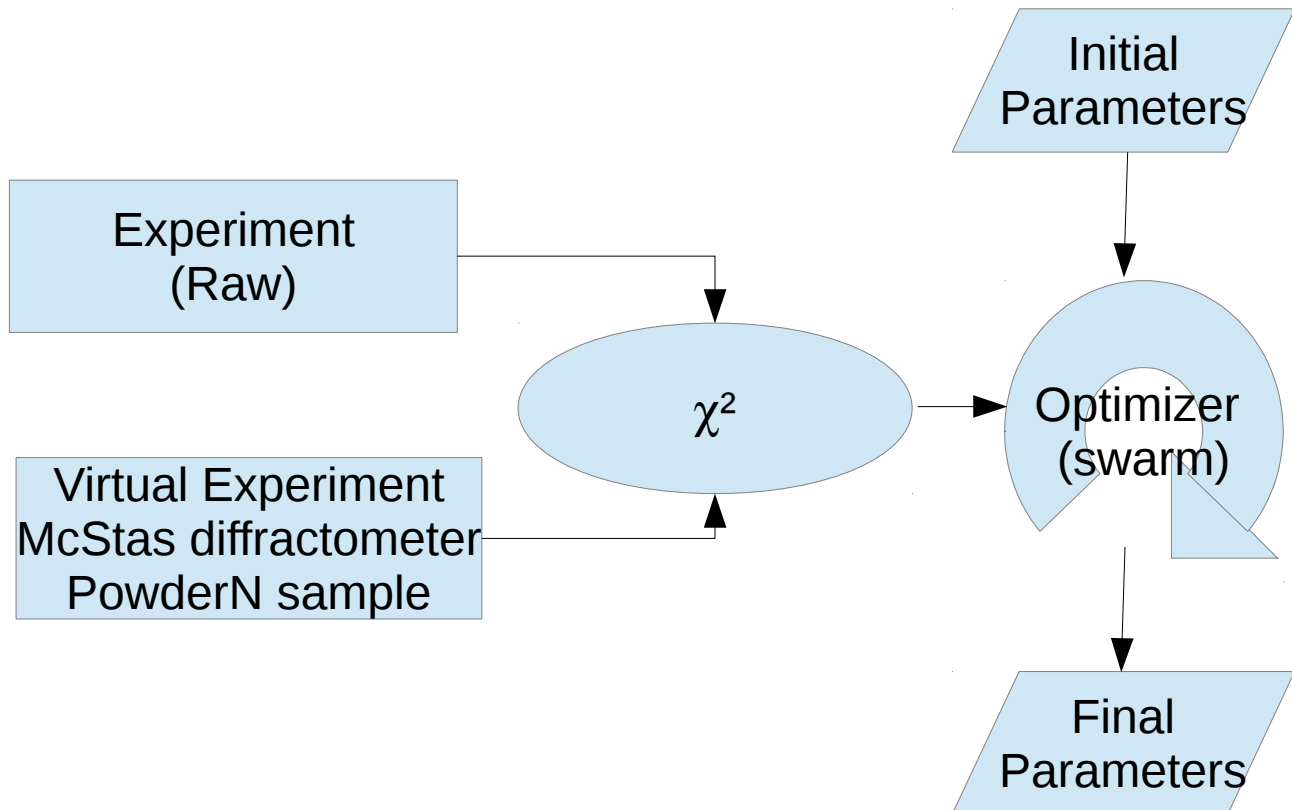
The powder refinement methodology, on a constant wavelength diffractometer, derives from the initial technique from H. Rietveld [14]. It consists in a least-square minimization of a model compared with the actual diffractogram. The model is built from a theoretical diffraction pattern for a perfect powder, convoluted with a peak shape depending on the characteristics of the beam, the experimental diffractometer arrangement, and the sample size and shape. In practice, the peak shape

is derived from a Gaussian or Voigt profile, possibly asymmetric, and the half width $H(\theta)$ at diffraction angle θ is inferred from the Caglioti formalism [15]:

$$H^2 = U \tan^2 \theta + V \tan \theta + W$$

where U, V and W depend on the horizontal beam collimation and monochromator horizontal mosaicity. In addition, a background line is added, usually as a polynomial function of θ with adjustable parameters. The perfect powder diffractogram (a set of Dirac peaks corresponding to reflections on lattice planes) depends on the sample structure, that is e.g. the lattice parameters, the space group, the atom positions in the cell, the occupancy, charge and thermal factor. Except for the sample parameters, most other refined parameters are phenomenological. This usually allows to account for imperfections of the instrument set-up.

In this section, we present a new Rietveld methodology which does not make use of explicit analytical approximations to model the instrument response. It consists in using a virtual experiment, with its configuration parameters, as a model compared to a measurement. An optimisation is then carried-out to determine the parameter set that result in the best match with the measurement. The schematic representation of this methodology is shown as Drawing 3. As a consequence, the measurement data set does not need to be reduced as this step is included intrinsically in the virtual experiment.



Drawing 3: Using a virtual experiment as a model for fitting a measurement data set.

3.3.1 The D2B diffractometer at the ILL

The diffractometer D2B is characterised by the very high take-off angle (135 deg) for the monochromator, which has a relatively large mosaic spread of 20' to compensate for the corresponding intensity ($d\lambda/\lambda$) loss. It is 300 mm high, focusing vertically onto about 50 mm; this large incident vertical divergence is matched by 200 mm high detectors and collimators. A complete diffraction pattern is obtained after about 100 steps of 0.025 deg in 2θ , since the 64 detectors are spaced at 2.5 deg intervals. Such scans take typically 30 minutes; they are repeated to improve statistics.

D2B was designed for work on samples and high resolution of very large d-spacings using wavelengths of between 2.4 Å and 6Å. Wavelengths can easily be changed under computer control, since they are all obtained by a simple rotation within the Ge[hhl] plane. A large graphite filter can be switched in to provide a very clean beam at 2.4 Å, and a cold Be-filter can be used for longer wavelengths.

The McStas model shown in Illustration 17 reproduces the geometry of the instrument, as installed at the ILL, and is included in the software distribution. Its main parameters are indicated in Table 4.

<u>Parameter</u>	<u>Description</u>
λ	Wavelength (Å)
Powder	The powder structure file, with $F^2(hkl)$ reflection list
$\alpha_1, \alpha_2, \alpha_3$	Collimator divergence (arc minutes)
ϵ	Monochromator horizontal mosaicity (arc minutes)
a,b,c, α, β, γ , space group	Sample lattice
(x,y,z, B_{iso} , occupancy, charge, spin) _i	Atom site parameters, i=1-N
Detector angular rotation	The detector rotation misalignment
Sample transversal shift	The sample lateral misalignment

Table 4: The D2B McStas instrument model parameters.

The monochromator is automatically focused to the sample position, with a curvature radius of 3 m. Its vertical mosaicity is set to 12 arc minutes.

The sample is modelled with the *PowderN* component [6] in the McStas instrument, surrounded by a Vanadium can. As the *PowderN* component requires a $F^2(hkl)$ reflection list, we generate this list from the structural information by making use of the CrysFML. This converter is included in the iFit distribution. The structural information can be given as a CIF, ShelX or CFL (FullProf) file.

ILL_D2B

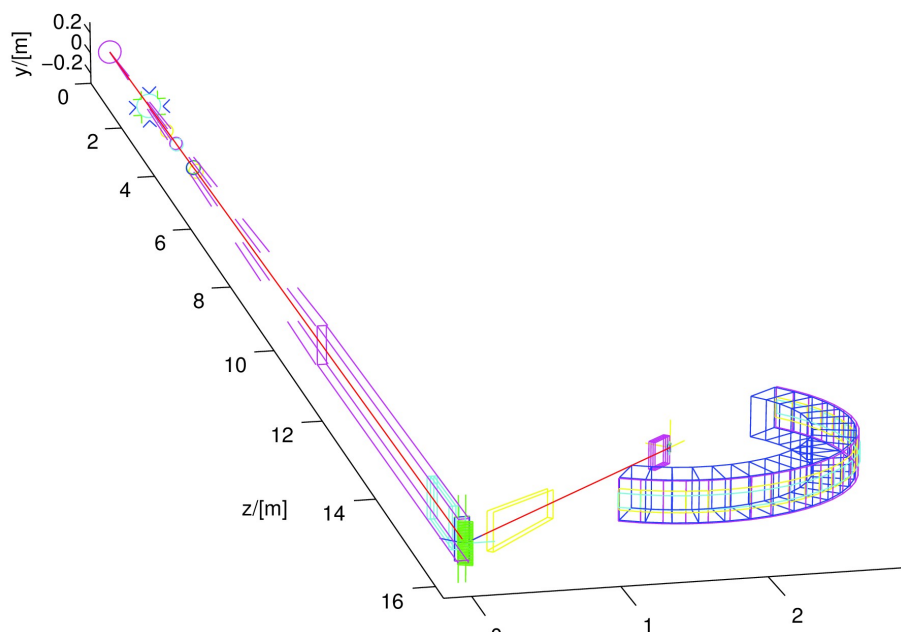


Illustration 17: The D2B diffractometer at the ILL, as modelled with McStas. The source is at the coordinate frame origin (left side), the multi-slab curved monochromator is at the end of the H11 beam tube (in green), the sample is at the centre of the radial collimator (in blue) in front of the detector.

Once launched with a powder description, the simulation produces both a full detector image (see Illustration 18), as well as its radial angular integration diffractogram (see Illustration 20). A computation of the powder diffractogram typically requires a few seconds.

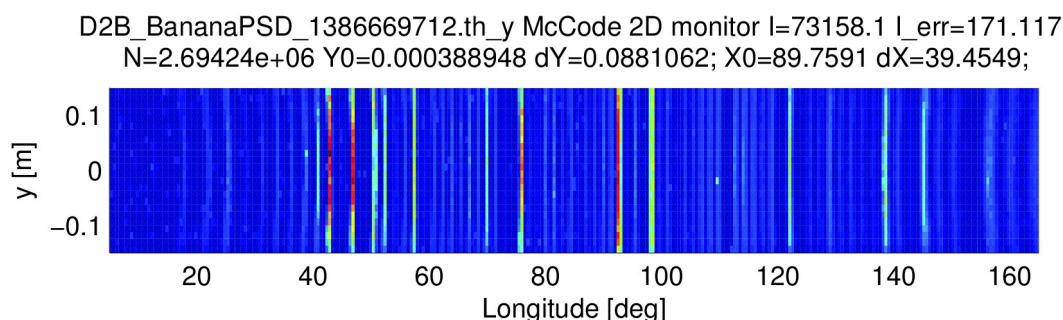


Illustration 18: The D2B detector image, simulated with McStas and a reference powder $\text{Na}_2\text{Ca}_3\text{Al}_2\text{F}_{14}$. The Debye-Scherrer rings are curved at low and high scattering angles, due to the detector height, and the peak width increases out of the equatorial plane, due to e.g. the monochromator curvature. The intensity is shown from blue (low) to red (high).

3.3.2 The powder refinement strategy

The instrument model with the powder sample takes structural and instrument parameters, and produces a diffractogram with raw detector counts. It thus behaves as a model function which can directly be used in a fitting routine to match a measured diffractogram. The default optimisation criteria is the least-square (but others are also available in iFit):

$$\chi^2 = \frac{1}{n-p-1} \sum (Signal - Model)^2 / Error^2$$

where n is the number of points in the diffractogram, and p is the number of free parameters to optimise. In this study, the *Signal* is the radially integrated diffractogram #50023 (2008) acquired on D2B, with its full detector coverage, with $n=3200$ points. The *Model* is the D2B instrument McStas description with the parameters listed in Table 4. The sample is a $\text{Na}_2\text{Ca}_3\text{Al}_2\text{F}_{14}$ powder. The *Error* is the square root of the measured counts. The Model contains 58 parameters, but some are kept fixed (α, β, γ , spins, occupancy and charges), some are constraint from symmetry ($b=c=a$, Ca along 'x', Al, Na and F3 along a 3-fold axis, F2 in the 'xy' plane). All free parameters are bound to limit the search space. A constant background (from the High Flux Reactor at the ILL) and a scaling factor are added. In the end, $p=25$ free parameters are optimised. The gradient based optimisers are not adapted to the virtual experiment model, which is intrinsically noisy from the random numbers implied by the Monte-Carlo technique. The iFit framework [11] provides a large choice of optimisation routines, which have been benchmarked for continuous and noisy problems. Among these, the heuristic particle swarm optimiser [58] was selected as it offers an excellent success ratio for noisy problems, at the cost of a long optimisation procedure.

The collimations and monochromator mosaicity have been left tunable within a short range around their nominal value, to account for possible imperfections of the instrument. In a similar way, a sample transversal shift and detector rotation misalignment have been added to the model.

With iFit, the refinement is triggered with the following few commands. First, the real measurement from D2B is imported with

```
>> measurement=iData('050023_full.dat');
```

Then the Rietveld model is assembled from the initial structure and the instrument description:

```
>> model = rietveld('ILL_D2B.instr', 'Na2Ca3Al2F14.cfl');
```

and the refinement is launched with

```
>> p = fits(measurement, model);
```

and produces the final refined parameter set p , including the instrument configuration and the fitted sample structure. In practice, a few additional lines are needed to add restraints (parameter bounds) and constraints (parameter cross dependencies).

3.3.3 The refinement results

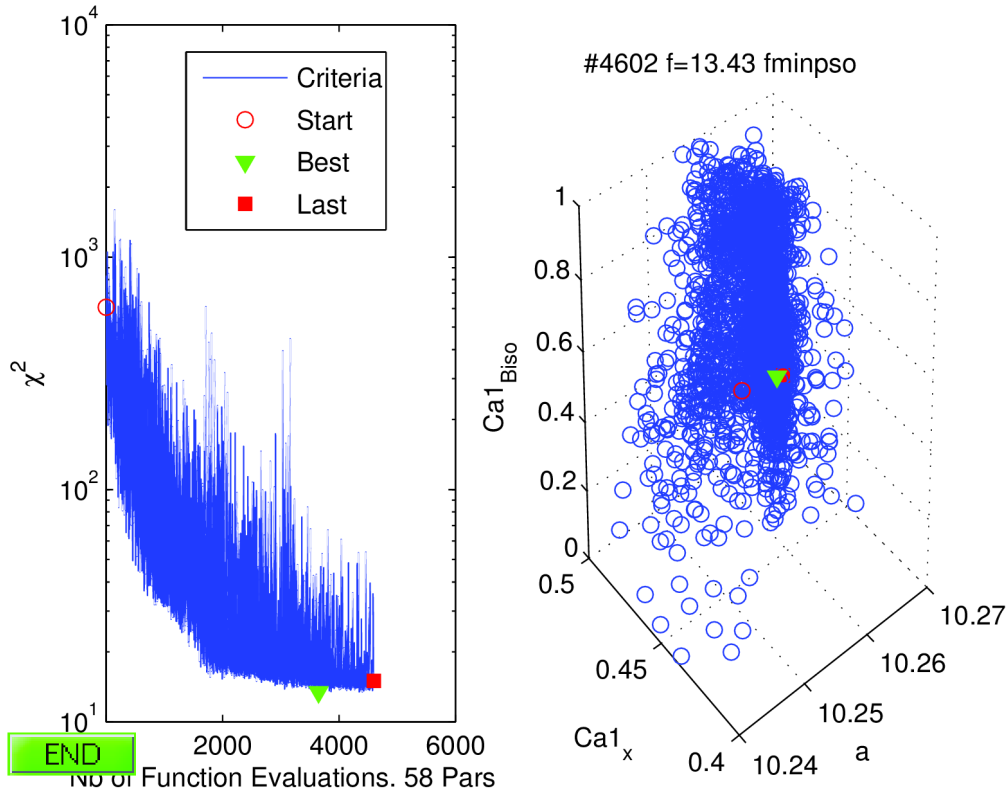


Illustration 19: The χ^2 optimisation during the Rietveld refinement procedure (left). The right plot shows the distribution of the first 3 varying parameters during the fit. This display is part of the iFit optimisation framework.

The refinement procedures requires about 2000 model evaluations to converge, as shown in the criteria evolution during the optimisation. We compute the final weighted profile and experimental R-factors:

$$R_{wp}^2 = \frac{\sum |Signal - Model|^2 / Error^2}{\sum |Signal|^2 / Error^2} \quad \text{and} \quad R_{exp}^2 = \frac{n - p}{\sum |Signal|^2 / Error^2}$$

as $R_{wp} = 0.150$ and $R_{exp} = 0.038$. The Pearson correlation coefficient is found as $r=0.891$. The final refined structural parameters are detailed in the Table 5. As seen in the final diffractogram in Illustration 20, the agreement with the measurement is good, even at low and large diffraction angles where the peak shapes are well accounted for. We remind that no analytical peak shape nor explicit convolution is used. Most structural parameters are in agreement with the published $\text{Na}_2\text{Ca}_3\text{Al}_2\text{F}_{14}$ structure [59], within the parameter uncertainties obtained from the parameter space local curvature analysis with a Monte-Carlo method. The lattice parameter is slightly above the expected value, but at the same time, the refined incoming neutron wavelength differs from its nominal value in the opposite direction.

<u><i>Parameter</i></u>	<u><i>Refined value</i></u>	<u><i>Published value [59]</i></u>
a=b=c	10.2612 ± 0.0012	10.2570
Ca1_x, y=0, z=1/4	0.4618 ± 0.0030	0.4667
Ca1_Biso	0.4434 ± 0.1595	
Al1_x=y=z	0.2478 ± 0.0094	0.2482
Al1_Biso	0.4169 ± 0.0646	
Na1_x=y=z	0.0831 ± 0.0039	0.0847
Na1_Biso	0.7717 ± 0.0495	
F1_x	0.1379 ± 0.0033	0.1387
F1_y	0.3035 ± 0.0029	0.3062
F1_z	0.1191 ± 0.0012	0.1206
F1_Biso	0.2304 ± 0.0245	
F2_x=y	0.3624 ± 0.0009	0.3630
F2_z	0.1881 ± 0.0032	0.1873
F2_Biso	0.4820 ± 0.0631	
F3_x=y=z	0.4619 ± 0.0023	0.4614
F3_Biso	0.1702 ± 0.0170	
Detector offset	0.0616° ± 0.0104°	0
Sample transversal shift	0.96 mm ± 0.15	0
α1	11.33' ± 3.64'	10'
α2	12.02' ± 0.62'	None (slit)
α3	7.26' ± 0.54'	5'
ε	17.78 ± 1.54	20'
Scaling factor	10.58 ± 0.36	
Background	368.5 ± 8.4	
λ	1.5961 ± 0.0002	1.6

Table 5: Refined structure of Na₂Ca₃Al₂F₁₄ and instrumental parameters from the virtual experiment powder refinement, compared to the published structure [59]. The parameter uncertainties are also indicated.

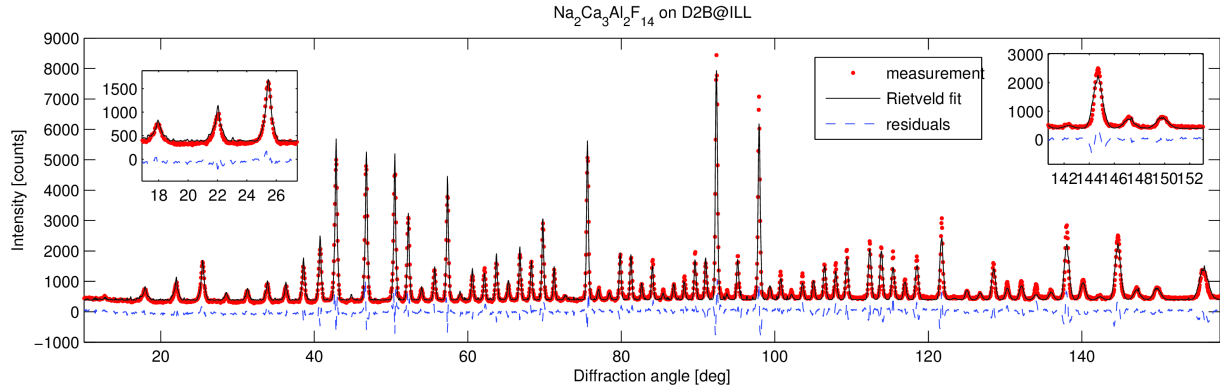


Illustration 20: The D2B $\text{Na}_2\text{Ca}_3\text{Al}_2\text{F}_{14}$ reference powder diffractogram measured, and refined using the virtual experiment as model. The final weighted profile R-factor is 0.15. Inserts show selected peaks at low and high diffraction angles.

The virtual experiment powder refinement is currently much slower than conventional Rietveld refinement as implemented in e.g. FullProf and CrysFML [17]. However, we have shown that this new methodology can provide similar results, without making any phenomenological assumption nor explicit analytical approximation. We thus believe it can efficiently complement conventional methods.

4 CONCLUSION AND OUTLOOK

We have presented a few usage examples of virtual experiments in view to analyse measurements. In all cases, the virtual experiment is shown to produce data which compares with that obtained on real instruments. This is possible when instrument models, as well as sample scattering kernels, are accurately described to match the real experiment configurations. The diversity of neutron optics and sample components available in McStas allows such advanced virtual experiments. Then, as presented in sections 3.1 and 3.2, the virtual experiment can be used to understand measurements, but also to demonstrate the feasibility of new experiments to motivate new measurements. In all cases, the virtual experiment produces raw data sets, which include intrinsically all usual measurements artefacts (resolution, detector, geometry, multiple scattering, transmission and self shielding, background, ...). The comparison with measurements can then be made on raw data, without explicit reduction procedure.

Moreover, such virtual experiments can be directly used as model to adjust its parameter to a measurement, as demonstrated in section 3.3. We shall discuss below the advantages and drawbacks associated to the use of virtual experiments, as well as the potential future developments of this technique.

4.1 Limitations of virtual experiment data analysis methodologies

The validity of the data analysis using a neutron scattering virtual experiment is intrinsically bound to the accuracy of the instrument and sample descriptions. The instrument is usually based upon

design drawings, from which deriving a McStas model is straightforward. However, neutron scattering instruments are continuously evolving, and the successive changes may be omitted in the instrument specifications. In this case, the instrument model will not match the actual instrument. In addition, instrument models contain ideal elements, as implemented in McStas. But all real instruments parts are manufactured and assembled within tolerance specifications.

In a conventional reduction/analysis methodology, these imperfections are often handled by phenomenological parameters. However, in a virtual experiment, all parameters correspond to explicit geometrical arrangements (distances, rotations, translations, shapes), or physical parameters – especially for neutron optics and samples. There are few tunable parameters, as most other quantities are fixed from the design specifications. A similar limitation applies for instance to *ab-initio* molecular dynamics, which in principle have no adjustable parameters as opposed to classical molecular dynamics where the potentials are fully parametrised. In the case of virtual experiments, it is possible to account for imperfections in the model for instance by including additional tilts, shifts, misalignments. For instance, the D2B model presented in section 3.3.1 includes a small offset rotation of the detector, and a sample lateral positioning shift. When the virtual experiment results do not match the actual measurements, one should then inquire about the sources of imperfections, and potentially missing components in the model. However, any virtual experiment should start from a simple model, which complexity is gradually increased to reach a realistic comparison level with real measurements.

The will to understand a measurement by comparing to the virtual experiment pushes the scientist to harness all kinds of systematic errors, and in the end benefits to the experiment by a better understanding of the response of the instrument.

4.2 What can be done to improve the applicability of virtual experiment data analysis

In order to cope with set-up imperfections, most real instruments require periodic alignment procedures. In practice, the misalignments still persist, but are often compensated by adjusting offsets. This refinement strategy corresponds to additional degrees of freedom, which may similarly be allowed in a virtual experiment. Tuning an ideal model to account for real imperfections takes time and effort.

In addition, a continuous development of sample kernels is needed, to account for increasing complexity in measured samples. The *PowderN* and *Isotropic_Sqw* scattering kernels can be used to model most materials. However, these components are perfect whereas real samples contains impurities, preferred orientations, domains, etc. Just as for the instrument models, the sample description should allow imperfections. In some cases, minor changes to the existing scattering kernels can improve the comparison of virtual experiment results with real measurements. But in many other cases, new in depth developments are required. For instance, McStas does not yet provide a convincing generic scattering kernel for elastic and inelastic processes in single crystals, which would also implement multiple scattering.

The computation time needed to evaluate a Monte-Carlo based virtual experiment with convincing statistics is certainly longer than that for equivalent analytical approaches. The Rietveld virtual

experiment based refinement example presented in section 3.3 is thousand times slower than a conventional methodology. A substantial work has thus to be carried out in terms of computational speed. The McStas models can efficiently run on clusters by making use of parallel processing (with MPI). The scaling is fully linear with the number of processing units and this has been benchmarked up to thousand processors. However, using such large high performance computing tasks always implies a significant time overhead (to e.g. start the tasks, send the software across nodes, send and receive information over the computing infrastructure). In practice, running on large clusters can only be envisaged when each computational step takes at least a few seconds, otherwise the infrastructure overhead becomes preponderant. In the case of a refinement procedure which uses heuristic optimisers such as the swarm strategy, many iterations are needed to reach convergence. To be competitive with conventional methods, the computation time of each virtual experiment iteration should be a fraction of a second, including potential infrastructure overhead. Using multi-threading has been tried with McStas, but as this is a pure Monte-Carlo code which accesses shared memory areas to store results, the appearance of dead-locks is unavoidable. An alternate solution to consider is to generalise the use of GPU cards, by e.g. using OpenCL, especially within scattering kernels.

4.3 Looking ahead

We believe that the use of virtual experiments for instrument design, experiment planning and understanding will become usual in the future. This is in particular essential for training of new users, and gaining knowledge on the capabilities of neutron scattering instruments. This training is already part of some schools and courses for students, to learn about neutron scattering. In addition, the use of virtual experiments should be generalised when designing new instruments so that the expected gains are evaluated not only on e.g. the integrated flux at the sample position, but with simulated measurements which exhibit elastic and inelastic features on top of instrumental artefacts.

As the computational efficiency of the virtual experiments will gradually be improved, their use within data analysis procedures will spread. Training on instrument models accuracy should be proposed, so that users can tune the virtual experiments to real set-ups, and actually make use of them in realistic comparisons with measurements, on raw data.

Extrapolating the presented Rietveld virtual experiment refinement, we could envisage other applications. For instance, entering a measured dynamic structure factor, which includes sample environment contribution, absorption and multiple scattering effects, the virtual experiment set-up could remove automatically the sample environment and multiple scattering contributions to extract a reduced data set. However, as mentioned in section 2.1, this procedure can not remove entirely the multiple-scattering contribution, which would require the knowledge of the dynamic structure factor on a doubled momentum range. Such an automatic treatment would still represent a breakthrough, as no generic multiple-scattering removal tool currently exists, and it would account accurately for the sample environment and absorption effects.

Applications of a data analysis based on virtual experiments could be proposed specifically for

- powder refinement, as presented in this manuscript, but made into a simple tool to use ;

- inelastic scattering, using e.g. $S(q, \omega)$ analytical models or a dynamic structure factor computed at each iteration from a classical molecular dynamics with adjustable potential parameters ;
- small angle scattering, with analytical $I(q)$ models ;
- single crystal diffraction structure refinement.

In our view, most of these aspects will certainly be accomplished within a decade, with usable computational speed.

5 References

- [1] J.F. Briesmeister, MCNP—A General Monte Carlo N-Particle Transport Code, version 4B, Report LA-12625-M, Los Alamos, 1997.
- [2] J. Šaroun, J. Kulda, *Physica B* **234-236** (1997) 1102-1104 ; see also <<http://neutron.ujf.cas.cz/restrax/>>
- [3] Wechslera, G Zsigmonda, F Streffera, J.A Strideb, F Mezei, *Physica B: Cond. Matt.* **276**, (2000) 71 ; see also <http://www.helmholtz-berlin.de/forschung/grossgeraete/neutronenstreuung/projekte/vitess/index_de.html>
- [4] K. Lefmann and K. Nielsen, *Neutron News* **10**, 20, (1999).
- [5] P. Willendrup, E. Farhi and K. Lefmann, *Physica B*, **350** (2004) 735.
- [6] Willendrup P, Filges U, Keller L, Farhi E, Lefmann K, “Validation of a realistic powder sample using data from DMC at PSI” , *Physica B-Cond. Matt.* **385** (2006) 1032.
- [7] E. Farhi, V. Hugouvieux, M.R. Johnson, W. Kob, “Virtual experiments: Combining realistic neutron scattering instrument and sample simulations“, *Journal of Computational Physics* **228** (2009) 5251.
- [8] M. Boin, *J. Appl. Cryst.* **45**, (2012). 603.
- [9] McStas web-site <<http://www.mcstas.org>>
- [10] E. Farhi and P. Willendrup, *Collection SFN* **12** (2011) 303-339 ; see also <<http://dx.doi.org/10.1051/sfn/201112015>>.
- [11] E. Farhi, Y. Debab and P. Willendrup, *J. Neut. Res.*, **17** (2013) 5. DOI: 10.3233/JNR-130001
- [12] iFit web site <<http://ifit.mccode.org>>.
- [13] D. Richard, M. Ferrand and G.J. Kearley, *J. Neutron Research* **4** (1996) 33-39 ; see also LAMP, ILL, France <http://www.ill.fr/data_treat/lamp/lamp.html>.
- [14] H. Rietveld, *J. Appl. Cryst.* **2** (1969) 65-71.
- [15] G. Caglioti, A. Paoletti, F.P. Ricci, *Nucl. Instr. and. Meth.* **3** (1958) 223
- [16] K. Lefmann; P. K. Willendrup; L. Udby; B. Lebeck; K. Mortensen; J. O. Birk; K. Klenø; E. Knudsen; P. Christiansen; J. Saroun; J. Kulda; U. Filges; M. Konnecke; P. Tregenna-Piggott; J. Peters; K. Lieutenant; G. Zsigmond; P. Bentley; E. Farhi *Journal of Neutron Research*, 1477-2655, Volume **16**, Issue 3, 2008, Pages 97 – 111.
- [17] J. Rodriguez-Carvajal, *Physica B* **192** (1993) 55.
- [18] G. Kresse and J. Hafner. “Ab initio molecular dynamics for liquid metals”. *Phys. Rev. B*, **47** (1993) 558.
- [19] Kresse G. , *J. Non-Cryst Sol.* **312** (2002) 52.
- [20] Kresse G and Furthmüller J, *Phys. Rev. B* **54** (1996) 1169-11186.
- [21] G R. Kneller, V Keiner, M Kneller and M Schiller, *Comp. Phys. Comm.*, vol. **91** (1995) 191-214.
- [22] V. Calandrini, E. Pellegrini, P. Calligari, K. Hinsén and G.R. Kneller, *Collection SFN* **12** (2011) 201-232 ; see also <<http://dx.doi.org/10.1051/sfn/201112010>> and <<https://forge.ill.eu/projects/nmoldyn>>.
- [23] T. Rog, K. Murzyn, K. Hinsén, G.R. Kneller, *J. Comput. Chem.* **24** (2003) 657.
- [24] N. Draper, M. Gygg, R. Taylor, M. Doucet, S. Campbell *et al*, Mantid <<http://mantidproject.org>>.
- [25] R.T. Azuah, L.R. Kneller, Y. Qiu, P.L.W. Tregenna-Piggott, C.M. Brown, J.R.D. Copley, and R.M. Dimeo, *J. Res. Natl. Inst. Stan. Technol.* **114** (2009) 341; see also <<http://www.ncnr.nist.gov/dave/index.html>>.
- [26] C. Dewhurst, GRASP, ILL, France <<http://www.ill.eu/instruments-support/instruments->

groups/groups/lss/grasp/home/>.

[27]J. Ollivier and J-M. Zanotti, *Collection SFN* **10** (2010) 379 ; see also <<http://dx.doi.org/10.1051/sfn/2010006>>.

[28]M. Johnson, N. Malikova and M. Plazanet, *Collection SFN* **10** (2010) 1-339 EDP Sciences ISBN:978-2-7598-0542-6.

[29]G.L. Squires, “*Introduction to the theory of thermal neutron scattering*”, Dover edition (1996) and Cambridge University Press (1978).

[30]V.F. Sears, *Neutron News* **3** (1992) 26.

[31]V.F. Sears. *Adv. Phys.*, **24**, (1975) 1.

[32]L.V. Maierov , *Soviet Atomic Energy* **19** (1955) 940.

[33]H.E. Fischer , A.C. Barnes and P.S. Salmon, *Rep. Prog. Phys.* **69** (2006) 233-299.

[34]S. W. Lovesey, *Theory of Neutron Scattering from Condensed Matter*, Vol. 1, Oxford Science Pub., Clarendon Press (1984), Oxford

[35]J.R.D. Copley, J.M. Rowe, *Phys. Rev. A* **9** (1974) 1656

[36]T. Bodensteiner, Chr. Morkel, and W. Gläser, *Phys. Rev. A* **45** (1992) 5709

[37]Data from experiment Test-1517 (2009) by J. Bossy *et al* on IN5 at the ILL with a l - ^4He cell and an incoming wavelength $\lambda=3$ Å, used to describe the He-II dynamic up to 7 meV energy transfer on the multi-phonon continuum (energy resolution 350 meV). T=1.7 K.

[38]Data from J. Bossy and H. Schober (2000) acquired on IN5 at the ILL with a l - ^4He cell and an incoming wavelength $\lambda=5$ Å, used to simulate the single phonon peak shape (resolution 100 meV). T=0.4 K.

[39]Data from the experiment 6-01-140 (1995) obtained by Gibbs *et al* on IN6 at the ILL with a l - ^4He cell and an incoming wavelength $\lambda=4$ Å covers most of the wavelength range used in the current experiment. T=1.3 K. An electronic artefact is noticeable at $q > 2.1$ Å $^{-1}$ for p =SVP,,5, 10, and 15 bar.

[40]H. N. Robkoff and R. B. Hallock , *Phys. Rev. B* **25** (1982) 1572.

[41]P. Schmidt-Wellenburg, K. H. Andersen, and O. Zimmer, *Nucl. Instr. Meth. Phys. Res. Sec. A* **611**, 259 (2009).

[42]F. Caupin, J. Boronat, K.H. Andersen, *J. Low Temp. Phys.* **152** (2008) 108.

[43]Baker et al, *Physics Letters A* **308** (2003) 67.

[44]W. Kohn, L. Sham, *Phys. Rev.* **140** (1965) A1113.

[45]J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.*, **77** (1996) 3865.

[46]P.E. Blöchl, *Phys. Rev. B* **50**, 17953 (1994).

[47]D. M. Ceperley and B. J. Alder, *Phys. Rev. Lett.* **45** (1980) 566–569

[48]G. Careri and A. Paoletti, *Il Nuovo. Cim.* **2** (1955) 574.

[49]Y. Waseda et al, *Z. Physik B* **32**, (1979) 253 .

[50]H. Ocken and C.N. Wagner, *Phys. Rev.* **149** (1966) 149.

[51]T. Scopigno and G. Ruocco, *Rev. Mod. Phys.* **77** (2005) 881

[53]H. Reichert, F. Bencivenga, B. Wehinger, M. Krisch, F. Sette, and H. Dosch1, *Phys. Rev. Lett.* **98** (2007) 096104.

[52]B. Wehinger, M. Krisch and H. Reichert , *New Journal of Physics* **13** (2011) 023021 .

[54]Y. Tsuchiya, *J. Phys. Soc. Jap.* **66** (1997) 120.

[55]O. J. Kleppa , *J. Chem. Phys.* **18** (1950) 1331.

[56]L. Hennet, I. Pozdnyakova, A. Bytchkov, V. Cristiglio, P. Palleau et al., *Rev. Sci. Instrum.* **77** (2006). 053903 .

[57]H. Totsmann et al, *Phys. Rev. B*, **59** (1999) 783

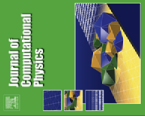
[58]Kennedy J., Eberhart R.C. (1995): Particle swarm optimization. In: *Proc. IEEE Conf. on Neural Networks*, IV, Piscataway, NJ, pp. 1942-1948

[59]G. Courbion, G. Ferey, *Journal of Solid State Chemistry* **76** (1988) 426.

6 Selected papers

E. Farhi, V. Hugouvieux, M.R. Johnson, W. Kob, “*Virtual experiments: Combining realistic neutron scattering instrument and sample simulations*“, *Journal of Computational Physics* **228** (2009) 5251.

E. Farhi, Y. Debab and P. Willendrup, *J. Neut. Res.*, **17** (2013) 5. DOI: 10.3233/JNR-130001.



Contents lists available at [ScienceDirect](#)

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Virtual experiments: Combining realistic neutron scattering instrument and sample simulations

E. Farhi^{a,*}, V. Hugouvieux^b, M.R. Johnson^a, W. Kob^c

^a Institut Laue Langevin, BP 156, 38042 Grenoble Cedex 9, France

^b INRA, UR1268 Biopolymères Interactions Assemblages, F-44300 Nantes, France

^c Laboratoire des Colloïdes, Verres et Nanomatériaux, Université Montpellier II, place E. Bataillon, 34095 Montpellier Cedex 5, France

ARTICLE INFO

Article history:

Received 19 June 2008

Received in revised form 26 March 2009

Accepted 8 April 2009

Available online 18 April 2009

ABSTRACT

A new sample component is presented for the Monte Carlo, ray-tracing program, McStas, which is widely used to simulate neutron scattering instruments. The new component allows the sample to be described by its material dynamic structure factor, which is separated into coherent and incoherent contributions. The effects of absorption and multiple scattering are treated and results from simulations and previous experiments are compared.

The sample component can also be used to treat any scattering material which may be close to the sample and therefore contaminates the total, measured signal.

© 2009 Elsevier Inc. All rights reserved.

PACS:

02.70.Ns

02.70.Uu

25.40.Dn

25.40.Fq

29.30.Hs

31.15.-p

47.11.Mn

61.05.fg

61.20.Ja

61.25.Mv

78.70.Nx

Keywords:

Neutron scattering

Molecular dynamics

Liquids

Instrument simulation

Monte Carlo simulation

Multiple scattering

1. Introduction

Neutron scattering provides a large variety of instruments to probe structure and dynamics of condensed matter. However, the technique is flux limited, which motivates continuous efforts to improve both the flux and the overall efficiency of the instruments. Therefore both analytical and numerical methods are used in order to determine optimal instrument configurations.

* Corresponding author. Tel.: +33 4 76 20 71 35; fax: +33 4 76 20 76 48.

E-mail address: farhi@ill.fr (E. Farhi).

For simple neutron beam configurations, analytical models are available to describe the different neutron optics elements of the instruments, which modify the characteristics of the beam (e.g. divergence, wavelength spread, spatial and time distributions). This approach is valid for individual elements such as guides [1,2], choppers [3,4], Fermi choppers [5,6], velocity selectors [7], monochromators [8–11], and detectors [12–14]. In the case of a limited number of optical elements, the so-called acceptance diagram theory [2,15,16] may be used, within which the neutron beam distributions are considered to be homogeneous, triangular or Gaussian. However, real neutron instruments are composed of a large number of optical elements, and this brings additional complexity by introducing strong correlations between neutron beam parameters like divergence and position – which is the basis of the acceptance diagram method – but also wavelength and time. The usual analytical methods, such as phase-space theory, then reach their limit of validity in the description of the resulting effects.

In order to cope with this difficulty, the simulation of neutron spectrometers may be performed using Monte Carlo methods (for a general review, see Ref. [17]), which are commonly used for the description of microscopic physical processes such as absorption, scattering or reflection. Integrating these events over the neutron trajectories results in an estimation of measurable quantities characterizing the neutron instrument. Moreover, using importance sampling reduces the computation time and gives better accuracy. Early implementations of the Monte Carlo method for neutron instruments used *home-made* computer programs (see [18,19]) but, more recently, general packages have been designed, providing models for most optical components of neutron spectrometers. The most widely-used packages are NISP [20], ResTrax [21], McStas [22], Vites [23], and IDEAS [24], which allow a wide range of neutron scattering instruments to be simulated. The neutron ray-tracing Monte Carlo method has been used widely for guide studies [15,25,26], instrument optimisation and design [27–29]. Since Monte Carlo ray-tracing gives accurate estimates for the flux and resolution, it enables optimum parameter sets for the instrument to be determined, which is not always possible with classical analytical methods. In addition, as for any instrument, in certain experiment configurations, the experimental signal may be contaminated due to various contributions to the total signal coming either from the instrument or from the sample such as background, coherent and incoherent scattering, self-shielding and multiple scattering.

Some of these questions may be tackled directly with Monte Carlo neutron scattering instrument simulations but others depend intrinsically on the sample. Indeed the experimental signal is the convolution of the instrument response and the signal due to the interaction between neutrons and the sample (structure and dynamics). Separating instrument and sample contributions requires both to be known in detail. As a consequence, the concept of virtual experiments [30], that is simulations including accurate models for both instruments and samples, has recently become a logical and important extension of Monte Carlo neutron scattering instrument simulations. For instance, for the multiple scattering contribution, no experimental method makes it possible to accurately measure this contribution, even though it can become significant at low q momentum transfers, for example, below the first diffraction maximum in liquids and glasses, where the single scattering coherent signal is weak in most materials. This is why attempts have been made to reduce the multiple scattering contribution by partitioning the sample with absorbing layers, as in [31]. However, this is not always applicable thus making the simulation approach very valuable.

Many methods and approximations have been developed to quantify these contributions and, for example, analytical formulae exist that can be applied for correcting multiple scattering [32]. However, these methods remain limited in their capability to handle strongly cross correlated neutron parameter states, which originate from, e.g. complex instrument and sample geometry descriptions. A number of previous Monte Carlo codes [33–36] have been designed to evaluate single and multiple scattering, absorption, self-absorption and transmission factors. However, they are often limited regarding, e.g. the instrument geometry, the sample environment and shape, or the type of experiment (diffraction, time-of-flight).

This paper presents the component *Isotropic_Sqw* in the *McStas* code [22]. It allows the sample scattering function $S(q, \omega)$, where q and ω are the wavevector and energy transfers, to be included in a Monte Carlo neutron scattering instrument simulation. This implies that both elastic and inelastic scattering are taken into account, for the coherent and incoherent processes. The object of the study is to demonstrate that complex instrument descriptions can be coupled to sample simulations in order to produce virtual experiment results that compare with real measurements.

In the next section, the general principle of sample simulations, dedicated to the global simulation of neutron scattering experiments, is explained. In Section 3 we compare a virtual experiment on liquid Rb with the direct analysis of the corresponding experimental data and we focus on the possibilities for handling multiple scattering in order to demonstrate the usefulness of such a numerical approach for the analysis of experimental data. While the rubidium work does not reveal a particular high level of multiple scattering, we regard the work of Copley [31] as a reference in this context. Finally we discuss the results and give some perspectives of the present work.

2. Sample simulation for virtual neutron scattering experiments

With a view to performing virtual neutron scattering experiments, a new sample component has been developed for the *McStas* package [22] 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 by this component). The component *Isotropic_Sqw* treats coherent and incoherent neutron scattering and may be used to model most materials, including sample environments with concentric geometries. The method presented here for handling neutron interaction with isotropic materials is similar in many respects to the earlier MSC [33], Discus [34] and MSCAT [35] methods, but the implementation presented here is part of a more general treatment of a sample in an instrument.

In the following, we consider an isotropic medium irradiated with a cold or thermal neutron beam. We ignore the possible thermal fission events and assume that the incoming neutron energy does not correspond to a Breit–Wigner resonance in the material. Furthermore, we do not take into account quantum effects in the material, nor refraction and primary extinction.

The justification of using the dynamic structure factor $S(q, \omega)$ for the description of the scattering events is given by following Squires [37]. The experimental counterpart of the scattering law $S(q, \omega)$ is the neutron double differential scattering cross section for both coherent and incoherent processes:

$$- \frac{d^2\sigma}{d\Omega dE_f} = \frac{\sigma}{4\pi} \frac{k_f}{k_i} NS(q, \omega), \quad (1)$$

which describes the number of neutrons scattered per unit solid angle $d\Omega$ and per unit final energy dE_f . In this equation, $N = \rho V$ is the number of atoms in the scattering volume V with atomic number density ρ . E_f , E_i , k_f , k_i are the kinetic energies and wavevectors of final and initial states respectively, σ is the bound atom scattering cross section, Ω is the solid angle and q, ω are the wavevector and energy transfer at the sample. In practice, the double differential cross section is a linear combination of the coherent and incoherent parts of the dynamic structure factor as:

$$\sigma S(q, \omega) = \sigma_{coh} S_{coh}(q, \omega) + \sigma_{inc} S_{inc}(q, \omega), \quad (2)$$

where σ_{coh} (resp. σ_{inc}) is the bound atom coherent (resp. incoherent) scattering cross section. Consequently the structure and dynamics of isotropic samples can be characterised by the quantity $\sigma S(q, \omega)$, which completely determines the interaction between neutrons and the sample and therefore can be used as a probability distribution of ω -energy and q -momentum transfers.

We consider a neutron with a given position and incident wavevector k_i and energy E_i . In view of computing the interaction between a neutron and a material of given volume and shape, the first step consists in determining the propagation path length in the material by geometrical intersections between the neutron trajectory and the sample volume. Along this path, the neutron may either interact with atoms of the sample, through absorption or scattering, or be transmitted without interaction with the material and exit the sample.

The probability of such an interaction is governed by the energy dependent total cross section σ_{tot} accounting for both scattering and absorption, defined as:

$$\sigma_{tot}(E_i) = \sigma_{abs}(E_i) + \sigma_s(E_i). \quad (3)$$

In this expression, the cross sections σ_{abs} and σ_s reflect the effective interaction surface for absorption and scattering respectively between a neutron of energy E_i and an atom of the material.

Therefore, before determining what happens to the neutron, we have to compute the different contributions to the total cross section, namely $\sigma_{abs}(E_i)$ and $\sigma_s(E_i)$. These quantities actually govern the ratio of absorption to scattering in the material for a given incident energy E_i .

2.1. Cross sections

Except for a few materials with absorption resonances in the cold-thermal energy range, the absorption cross section for an incoming neutron of velocity $v_i = \sqrt{2E_i/m}$ (in m/s), where m is the neutron mass, is computed as

$$\sigma_{abs}(E_i) = \sigma_{abs}^{2200} \frac{2200}{\sqrt{2E_i/m}}, \quad (4)$$

where σ_{abs}^{2200} is the absorption cross section for a neutron with $v_i = 2200$ m/s and is obtained from the literature [38].

Following Sears [32], the total scattering cross section for incoming neutrons with initial energy E_i is

$$\sigma_s(E_i) = \int \int \frac{d^2\sigma}{d\Omega dE_f} d\Omega dE_f = \frac{N\sigma}{4\pi} \int \int \frac{k_f}{k_i} S(q, \omega) d\Omega dE_f, \quad (5)$$

where the integration runs over the entire space and all final neutron energies. As the dynamic structure factor is defined in the (q, ω) space, the integration requires a variable change. Using the momentum conservation law $\vec{q} = \vec{k}_i - \vec{k}_f$ and the solid angle relation $\Omega = 2\pi(1 - \cos\theta)$, where θ is the solid angle opening, we obtain:

$$\sigma_s(E_i) = N \int \int \frac{\sigma S(q, \omega) q}{2k_i^2} dq d\omega. \quad (6)$$

This integration runs over the whole accessible (q, ω) dynamical range for each incoming neutron. The dynamic structure factor needs only to be known for $\omega > 0$ (anti-Stokes processes, neutron gains energy), as

$$S(q, -\omega) = e^{h\omega/k_B T} S(q, \omega). \quad (7)$$

This correction will be applied in Section 2.5, and we now consider that $\omega > 0$. In practice, the knowledge of the dynamic structure factor is defined over a limited area with $q \in [q_{min}, q_{max}]$ and $\omega \in [\omega_{min}, \omega_{max}]$ which is constrained by the method for obtaining $S(q, \omega)$, i.e. from previous experiments, molecular dynamics simulations, and analytical models. It is desirable that this area be as large as possible, starting from 0 for both ranges. If we use $\omega_{min} \rightarrow 0$, $q_{min} \rightarrow 0$, $h\omega_{max} > 4E_i$ and $q_{max} > 2k_i$,

we completely describe all scattering processes for incoming neutrons with wavevector k_i [33]. This means that in order to correctly estimate the total intensity and multiple scattering, the knowledge of $S(q, \omega)$ must be wider (at least twice in q , as stated previously) than the measurable range in the corresponding experiment. As a side effect, a self-consistent iterative method for finding the true scattering law from the measurement itself is not theoretically feasible, except for providing crude approximations. However, the measured dynamic structure factor may be used to estimate the multiple scattering for a further measurement using at least twice longer wavelength neutrons. Extrapolating the scattering law beyond the accessible measurement range might improve substantially the applicability of the method, but such a discussion is beyond the scope of this paper.

Consequently, limiting the q integration in Eq. (6) to the maximum momentum transfer for elastic processes $2k_i$, we write the total scattering cross section as

$$\sigma_s(E_i) \simeq \frac{N}{2k_i^2} \int_0^{2k_i} q \sigma_S(q) dq. \quad (8)$$

Using Eq. (2), it is possible to define similar expressions for the coherent and incoherent terms $\sigma_{coh}(E_i)$ and $\sigma_{inc}(E_i)$ respectively. These integrated cross sections are usually quite different from the tabulated values [38] since the latter are bound scattering cross sections.

Once the absorption and scattering cross sections are known, we can compute transmission, absorption and scattering event probabilities.

2.2. Transmission

The transmission probability follows an exponential decay law accounting for the total cross section. The neutron trajectory intersection with the sample geometry provides the total path length in the sample d_{exit} to the exit. Defining the linear attenuation $\mu(E_i) = \rho \sigma_{tot}(E_i)$, the probability that the neutron event is transmitted along path d_{exit} is $e^{-\mu(E_i)d_{exit}}$. Consequently, we determine if the neutron interacts with the sample by drawing a random number ξ_i in the range $[0, 1]$. Note that whenever we refer to random numbers, a uniform generator is used [39]. In the following, we introduce the neutron Monte-Carlo statistical weight, which represents the probability of a given neutron event to participate in the total intensity at a given position in the course of the simulation.

In case of transmission, the neutron leaves the sample. In previous Monte Carlo codes such as MSC [33], DISCUSS [34] and MSCAT [35], each exiting neutron event is forced to scatter to the detector area in order to improve the sample scattering simulation statistics and reduce the computing time. This method requires to focus the measurable energy and momentum transfers for the last scattering event in the sample towards the detectable area, so that the choice of the scattering excitation actually requires a more complex sampling mechanism from the dynamic structure factor. The corresponding instrument model is limited to a neutron event source, a sample and a detector. In the current implementation, it is equally possible to 'force' neutron events to scatter (and possibly reach the detector) by applying a correction factor $\pi_0 = 1 - e^{-\mu(E_i)d_{exit}}$ to the neutron statistical weight, in which case there is no need to cast the random number ξ_i for transmission probability. However, the *McStas* instrument model is often built from a large sequence of components [22]. Even though the instrument description starts as well with a neutron event source, more than one sample may be encountered in the course of the neutron propagation and multiple detectors may be positioned anywhere in space, as well as other instrument components (e.g. filters, mechanical parts, samples, shields, radiation protections). This implies that, in this case, neutron events scattered from a sample volume should not focus to a single area. Indeed, transmitted events may reach other scattering materials and it is not desirable to force all neutron events to scatter. In this case, the correction factor π_0 is therefore not applied, and neutron events can be transmitted through the sample volume. The simulation efficiency for the scattering is certainly lower, but enables much more complex arrangements, such as concentric sample environments, magnets and monochromator mechanical parts, and neutron filters, to be modelled.

2.3. Scattering and absorption

If the neutron is not transmitted, it may be either absorbed or scattered. In order to avoid losing neutron events through absorption, we consider that all interacting neutron events scatter and the neutron statistical weight is multiplied by a factor

$$\pi_1 = \frac{\sigma_s(E_i)}{\sigma_{tot}(E_i)} \quad (9)$$

to account for the fraction of absorbed neutrons along the path. Additionally, the type of interaction (coherent or incoherent) is chosen randomly with fractions $\sigma_{coh}(E_i)/\sigma_s(E_i)$ and $\sigma_{inc}(E_i)/\sigma_s(E_i)$.

Then we select a scattering position along the path, taking into account the secondary extinction and absorption probability. In this process, the neutron is considered to be a particle or an attenuated wave. The position of the neutron scattering event along the neutron trajectory length d_{exit} is determined by [19,34]

$$d_s = -\frac{1}{\mu(E_i)} \ln(1 - \xi_d [1 - e^{-\mu(E_i)d_{exit}}]), \quad (10)$$

where ξ_d is a random number in $[0, 1]$. This expression takes into account secondary extinction, originating from the decrease of the beam intensity through the sample (self-shielding).

2.4. Choice of the energy and momentum transfers

Once a scattering position has been assigned, the neutron interacts with a material excitation. Here we turn to the wave description of the neutron, which interacts with the whole sample volume. The distribution of excitations, which determines their relative intensity in the scattered beam, is simply the dynamic structure factor – or scattering law – $S(q, \omega)$. We build probability distributions from the scattering law in order to improve the efficiency of the method by favoring the (q, ω) regions with larger $S(q, \omega)$ values.

The choice of the (q, ω) wavevector-energy transfer pair could be done randomly, as in the first event of the second order scattering evaluation in DISCUS [34], but it is somewhat inefficient except for materials showing a broad quasi-elastic signal. As the scattering originates from structural peaks and excitations in the material $S(q, \omega)$, it is usual [35] to adopt an importance sampling scheme by focusing the (q, ω) choice to areas where the intensity of $S(q, \omega)$ is high. In practice, this means that the neutron event should scatter preferably on, for example, Bragg peaks, quasielastic contributions and phonons.

The main idea to implement the scattering from $S(q, \omega)$ is to cast two consecutive Monte Carlo choices, using probability distribution built uniformly from the dynamic structure factor. We define first the probability $P_\omega(\omega)d\omega$ as the *unweighted* fraction of modes whose energy lies between ω and $\omega + d\omega$

$$P_\omega(\omega) = \frac{\int_0^{\theta_{\max}} S(q, \omega) dq}{|\mathcal{S}|}, \quad (11)$$

where $|\mathcal{S}| = \int \int S(q, \omega) dq d\omega$ is the norm of $S(q, \omega)$ in the available dynamical range $q \in [q_{\min}, q_{\max}]$ and $\omega \in [\omega_{\min}, \omega_{\max}]$. The probability P_ω is normalised to unity, $\int_{\omega_{\min}}^{\omega_{\max}} P_\omega(\omega') d\omega' = 1$, and is a probability distribution of mode energies in the material. The energy transfer ω for scattering is determined by casting a random number $\xi_\omega \in [0, 1]$ and solving the equation

$$\xi_\omega = \int_{\omega_{\min}}^{\omega} P_\omega(\omega') d\omega'. \quad (12)$$

Similarly, in order to focus the wavevector transfer choice, we define the probability distribution of wavevector $P_q(q | \omega) d\omega dq$ for the selected energy transfer lying between ω and $\omega + d\omega$

$$P_q(q | \omega) = \frac{S(q, \omega)}{S(q)}, \quad (13)$$

from which we choose randomly a wavevector transfer q , knowing the energy transfer ω . As for the energy transfer, we cast a random number $\xi_q \in [0, 1]$ and determine the corresponding wavevector transfer q which solves the equation

$$\xi_q = \int_{q_{\min}}^q P_q(q' | \omega) dq'. \quad (14)$$

The right members in Eqs. (12) and (14) are monotonically increasing primitives of P_ω and P_q , which only depend on the dynamical structure factor S . This procedure ensures that the energy and wavevector choice is performed on a uniform (q, ω) space but statistically focuses scattering events where the scattering function is higher.

Then a selection between energy gain and loss is performed with the detailed balance ratio $e^{-h\omega/k_B T}$. In the case of Stokes processes, the neutron can not lose more than its own energy to the sample dynamics, so that $h\omega < E_i$. This condition breaks the symmetry between up-scattering and down-scattering.

2.5. Solving selection rules and choosing the scattered wavevector

The next step is to check that the conservation laws

$$h\omega = E_i - E_f = \frac{h^2}{2m} (k_i^2 - k_f^2), \quad (15)$$

$$\vec{q} = \vec{k}_i - \vec{k}_f \quad (16)$$

can be satisfied. These conditions are closely related to the method for selecting the outgoing wavevector direction.

When the final wavevector has to be computed, the quantities k_i , $h\omega$ and $q = |\vec{q}|$ are known. We solve the energy conservation law Eq. (15) and we select randomly k_f as one of the two roots. The scattering angle θ from the initial k_i direction is determined from the momentum conservation law $\cos(\theta) = (k_i^2 + k_f^2 - q^2)/(2k_i k_f)$, which defines a scattering cone. We then choose randomly a direction on the cone. If the selection rules can not be verified (namely $|\cos(\theta)| > 1$), a new (q, ω) random choice is performed (see Section 2.4). It might appear inefficient to select the energy and momentum transfers first and check the selection rules afterwards. However, in practice, the number of iterations to actually scatter on a high probability process and satisfy these rules is limited, usually below 10. Moreover, as these two steps are simple, the whole process requires a limited number of computer operations.

Once the scattering probability and position, the energy and momentum transfers and the neutron momentum after scattering have all been defined, the whole process is iterated until the neutron is transmitted and exits the sample volume.

2.6. Schematic neutron-matter interaction implementation

The processing of the interactions between neutrons and sample can be summarized as follows:

- (1) Compute the propagation path length in the material by geometrical intersections between the neutron trajectory and the sample volume.
- (2) Evaluate the total cross section from the integration of the scattering law over the accessible dynamical range.
- (3) Use the total cross section to determine the probability of interaction for each neutron along the path length, and select a scattering position.
- (4) Weight neutron interaction with the absorption probability and select the type of interaction (coherent or incoherent).
- (5) Select the wavevector and energy transfer from the dynamic structure factor used as a probability distribution. Apply the detailed balance.
- (6) Check whether selection rules can be solved. If they cannot, repeat (5).
- (7) Iterate this procedure from step 1 to 6 until the neutron leaves the sample.

The *McStas Isotropic_SqW* component implementation requires to provide the coherent and incoherent dynamical structure factor as two text files containing a matrix of $S(q, \omega)$ values on specified momentum and energy axes. Additionally, bound cross sections, temperature, density of material and molecular weight can be specified within the file headers, or given as individual parameters to the component. The sample geometry is described from a set of simple shapes (box, cylinder, sphere), or using a file containing a set of point coordinates to define a closed volume (which may for instance be obtained by scanning a real sample with a laser probe).

As the component is part of a full instrument description, the incoming neutron beam is obtained from *McStas* components upwards in the instrument description, which usually include a neutron source (pulsed or continuous) and neutron optics (guides, choppers, monochromators, ...). Any sample container is treated just as the sample itself, i.e. as a material surrounding the sample to be studied. This implies the ability of the component to handle embedded geometries as well as multiple scattering between the container and the sample, which are both implemented in the component code and *McStas*. The results of the computation are obtained using detector components following the sample component position in the instrument simulation sequence such as banana shaped monitors recording time and angular histograms (tallies). Some of these monitors can be made sensitive to, e.g. only multiple scattering, or scattering taking place in the container. Each histogram is saved into a single text file. There is no assumption in the component *Isotropic_SqW* regarding the following detector specifications (type, dimensions).

We shall now present an example of virtual experiment using this method, comparing simulated results with actual experiments.

3. A virtual experiment: liquid Rb on a time-of-flight spectrometer

All classes of instruments have been simulated with *McStas* and some of these are included as examples in the software [22]. To date, these simulations have used very simple models to describe samples, for example a vanadium-like constant incoherent scatterer, elastically scattering powders and a simple inelastic scatterer, but without multiple scattering.

In order to use the sample model presented above, prior knowledge of the dynamic structure factor $S(q, \omega)$ is required. Currently, there are three methods to obtain the dynamic structure factor (q, ω) matrix required for the sample simulation:

- (1) Use an analytical $S(q, \omega)$ function to generate a matrix with a regular q and ω sampling. Results obviously depend on the analytical models used. This is one of the methods used in Discus [34] and MSCAT [35].
- (2) Use previous experimental data, taking into account detector efficiencies, instrument contribution, sample absorption, empty cell and background subtraction. The resulting dynamical structure factor depends on the quality of the data analysis. Moreover, the extracted $S(q, \omega)$ should only be used to simulate experiments on half of its dynamical range (see comment in Section 2.1).
- (3) Compute the dynamic structure factor of the sample from molecular dynamics (MD).

In order to demonstrate the efficiency of the simulation methodology presented above, we have searched for past experiment results published together with estimates of multiple scattering effects. The extensive experiments on *l*-Rb by Copley et al. [31,40] and Demmel et al. [41,42] gave a complete dynamic structure factor data set, and additionally estimate the multiple scattering effects using MSCAT [35]. Moreover liquid rubidium is a simple alkali liquid which can be accurately simulated by means of classical [41,43] and *ab initio* molecular dynamics methods. We have thus chosen the published liquid rubidium dynamics measurements as a basis for the validation of our code. Complementary virtual experiment results on other materials will be published separately.

In the following, we have set up a simple model of an instrument built with a source, a liquid sample and a large detector.

3.1. Simulation details

The instrument model mimics the beam characteristics of the hybrid time-of-flight spectrometer built at the Argonne CP-5 reactor [44]. The neutron beam is extracted from a continuous thermal source ($T = 300$ K), collimated in a vacuum tube with width and height of 4 cm, and monochromatized to produce a $E_0 = 4.94$ and 33.0 meV incoming energy (we use here the same notation as in Copley paper, $E_0 = E_i$) with an energy resolution of $\Delta E = 0.24$ and 1.07 meV full-width at half maximum (FWHM) respectively. A Fermi Chopper creates a pulse of neutrons with triangular time distribution of 6.4 and 9.5 μ s half width for $E_0 = 4.94$ and 33 meV incoming energy respectively. The beam hits the sample at 2.88 m after being monochromatized. The incoming neutron flux (i.e. the reactor power) at the sample position was adjusted to match the intensity measurements reported in [44] as 1.5×10^8 and 1.7×10^8 neutrons/s for $E_0 = 4.94$ and 33.0 meV incoming neutron energies respectively. An oscillating radial collimator is positioned around the sample chamber in order to remove most of the furnace aluminum scattering contribution. The detector bank, centered on the sample position, has a radius of 2.5 m, with 205 tubes of height 45 cm covering an angular range from 10° to 120° . Each detector tube is filled with an ^3He gas at 6 bars and a stopping gas. The whole detector produces an angle-time intensity histogram. The simulation model only considers a single neutron pulse, and thus neglects frame overlap. The γ -ray background from the neutron source is neglected, as well as scattering in air and helium gas.

The simulated liquid rubidium sample geometry is as described in [31]. Disks of height and diameter 1.68 cm are stacked, separated by an infinitely thin absorbing material. The cylindrical assembly is contained in an aluminum can of thickness 0.045 cm. The sample cell total height is larger than the beam size. The sample is placed in an aluminum evacuated tank of diameter 30 cm, thickness 2 mm. The aluminum container elastic coherent scattering is modelled [45] using measured structure form factors [46], with constant $S_{\text{inc}}(q) = 1$ incoherent contribution.

The rubidium sample structure and dynamics are extracted from a classical molecular dynamics (MD) simulation [30]. The inter-atomic interaction is described by the effective two-body potential of Kambayashi and Kahl [47,48], with a core radius parameter $r_c = 1.307$ Å. This simple potential has been validated for temperatures close to the l -Rb melting point $T_m = 312$ K. In this study, we have used 520 rubidium atoms with a number density $n = 0.010288$ atoms per Å³. The system was first equilibrated at $T = 315$ K in the (N, V, T) ensemble, and trajectories were then recorded over 50,000 time steps of $\Delta t = 1$ fs in the (N, V, E) microcanonical ensemble. The sound velocity is obtained from the small- q phonon slope at about $c = 1300$ m/s and the diffusion coefficient is determined from the molecular dynamics as $D = 0.23$ Å²/ps at $T = 315$ K, in agreement with [43].

From the molecular dynamics simulation trajectories, we calculate the liquid rubidium dynamic structure factor $S(q, \omega)$ describing the structural and dynamical behaviour of the sample. $S(q, \omega)$ is computed with the help of the *nMoldyn* package [49], which allows us to evaluate separately the coherent and incoherent parts of the dynamic structure factor. We then use that data as input for the *McStas* [22] virtual experiment, using the instrument described above.

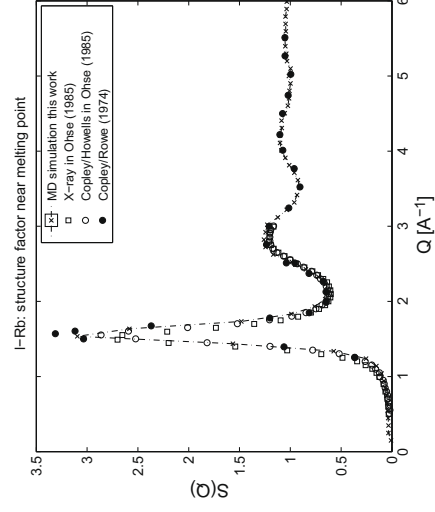


Fig. 1. Structure factor of l -Rb near the melting point: neutron scattering experiments [31,50] (circles), X-ray scattering [50] (squares) and MD simulation (crosses) at $T = 315$ K.

3.2. Results

In order to check the liquid rubidium molecular dynamics model, we first compare in Fig. 1 the static structure factor $S(q) = \int S(q, \omega) d\omega$ (where integration runs over the largest possible energy range) obtained from Copley measurements [31,50], X-ray data [50] and from our MD simulation. The simulated data are in good agreement with measurements, both in amplitude and phase. Around the first sharp peak at $q \sim 1.5 \text{ \AA}^{-1}$ simulation data is shifted to slightly lower q -values and is slightly sharper. This may originate from the limited size of the simulation box, which tends to slightly order the sample and thus show stronger structure peaks. As a consequence, we should expect in our simulation a stronger and sharper elastic peak contribution. These structure results are also in good agreement with those of Demmel [42]. This level of agreement is important to demonstrate since most of the multiple scattering events correspond to elastic scattering processes.

Copley [31] has measured the scattering function of liquid rubidium as a function of the time-of-flight and scattering angle. In order to determine the scattering function S from the measurement, the container and the instrument effects were subtracted and multiple scattering was estimated using MSCAT [35]. Such a data analysis procedure depends on the analytical and numerical models used.

In the current virtual experiment, the container and instrument are part of the simulation, and it is thus possible to produce raw data sets as if measured directly during the experiment, including the scattering from the sample as well as other contributions arising from the multiple scattering, the instrument geometry and the sample environment. In order to demonstrate how virtual experiments can produce results similar to the experimental ones, we have chosen to show the simulated raw time-of-flight data as a function of the angle, without any additional data analysis.

We have performed two simulations with incoming energies $E_0 = 4.94 \text{ meV}$ and $E_0 = 33 \text{ meV}$. Each simulation generated 10^8 initial neutron events and ran for less than 15 min on a single processor machine. The simulation results correspond to

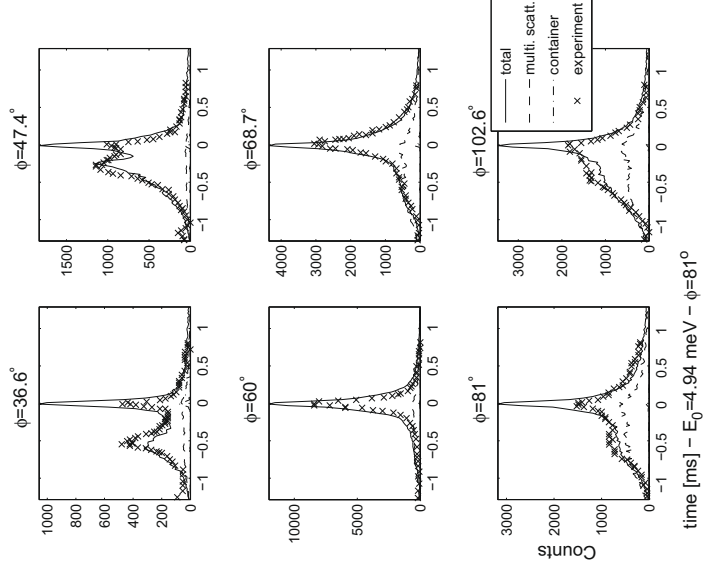


Fig. 2. I-Rb scattering (raw data) with incoming flux at $E_0 = 4.94 \text{ meV}$ showing the simulated total signal from sample (line) compared with the experimental data (crosses) from Ref. [31]. The sample environment (dash-dotted line, low intensity) and multiple scattering (dashed line) simulated contribution are also shown. The multiple scattering contribution has been multiplied by a factor 10, except for the two lowest angles. The corresponding wavevector transfer values for angles $\phi = 36.6^\circ, 47.4^\circ, 60^\circ, 68.7^\circ, 81^\circ$ and 102.6° are $q = 1.06, 1.22, 1.37, 1.5, 1.68$ and $1.76 \text{ \AA}^{-1}$ on the elastic line respectively.

an equivalent measurement time of the order of 40 h, but the corresponding experimental counting time is not indicated in [31]. Results of the virtual experiments using single detector tubes are shown in Figs. 2 and 3 and compared to time-of-flight experimental data from Ref. [31] at selected scattering angles.

It can be seen that both measurements and simulation show identical line shapes, with a sharper elastic line in the case of the simulation, as expected from the simulated structure factor (Fig. 1). Low angle results for $E_0 = 4.94$ meV incoming energy clearly present a distinct peak aside the elastic line. This peak, after transforming the time axis into energy transfer, reveals the liquid rubidium damped phonon in the case of the two lowest detection angles (see Fig. 4). However, this is not the case for the larger angles, in which case the apparent side peak does not correspond to any coherent single excitation, but originates from the time distribution transformation into energy bins, and disappears when extracting $S(q, \omega)$ [31]. The $E_0 = 33$ meV thermal incident energy results show a quasi-elastic line, slightly asymmetric.

The scattering from the sample container and the evacuated tank has also been simulated and appears as a central sharp peak, slightly displaced in time due to the cylindrical geometry. The multiple scattering contribution appears as a large quasi-elastic line which does not exceed a few percent of the total scattering. All contributions are obtained simultaneously, without additional scaling.

In order to compare our multiple scattering estimates with previous estimates like MSCAT [35], we have selected in Fig. 4 two representative plots of the scattering function at constant angle. The simulated data is obtained by changing the angle-time detector into an angle-energy transfer detector. This is achieved by storing at the sample position the energy transfer value, and access that information when neutrons reach the detector surface. There is no data analysis transformation and the result is obtained directly from the simulation, and not by treating the angle-time data as one would do from a real experiment. For the $E_0 = 4.94$ meV configuration, the lowest angle $\phi = 36.6^\circ$ shows the I -Rb phonon contribution, as well as the multiple scattering calculation. For the $E_0 = 33$ meV, we present results at scattering angle $\phi = 51^\circ$. As in the raw data, our simulated elastic line is sharper than the measured data. Both the scattering function and the multiple scattering contribution are in good agreement with the experiment and the MSCAT computation [31].

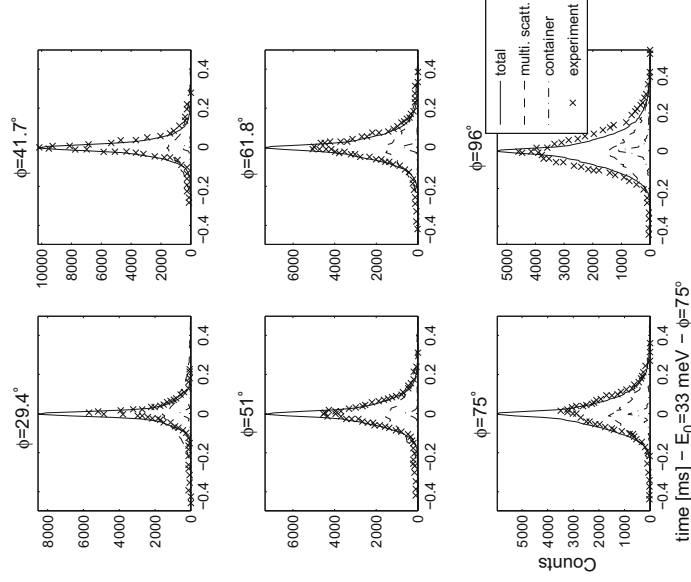


Fig. 3. I -Rb scattering (raw data) with incoming flux at $E_0 = 33$ meV showing the simulated total signal from sample (line) compared with the experimental data (crosses) from Ref. [31]. The sample environment (dash-dotted line) and multiple scattering (dashed line) simulated contribution are also shown. The multiple scattering contribution has been multiplied by a factor 10. The corresponding wavevector transfer values for angles $\phi = 29.4^\circ, 41.7^\circ, 51^\circ, 61.8^\circ, 75^\circ$, and 96° are $q = 2.265, 3.1, 3.45, 3.76$, and $3.16 \text{ \AA}^{-1}$ on the elastic line respectively.

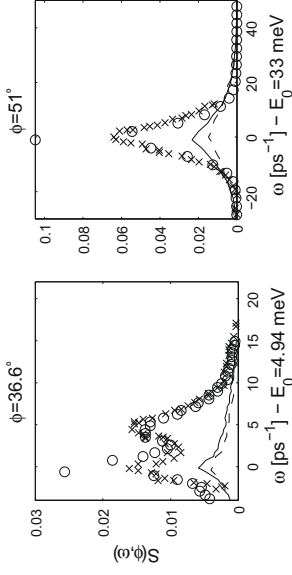


Fig. 4. I-Rb scattering function at constant angle with incoming flux $E_0 = 4.94$ meV (left) and $E_0 = 33$ meV (right) showing the simulated total signal from sample (circles) compared with the experimental data (crosses) from Ref. [31], and multiple scattering simulated in this work (line) and with MSCAT (dotted line). The corresponding wavevector transfer values for angles $\phi = 36.6^\circ$ and 51° are $q = 1.06$ and $3.1 \text{ \AA}^{-1}$ on the elastic line respectively. The multiple scattering contribution is multiplied by a factor 10 for the $E_0 = 33$ meV configuration (right).

The sample is found to scatter 7.8% and 10% of the incoming intensity and the multiple scattering represents 5.4% and 4.8% of the scattering intensity at $E_0 = 4.94$ meV and $E_0 = 33$ meV incident energy respectively. The absorption fraction, including self-shielding, represents 11.1% and 4.5% at $E_0 = 4.94$ meV and $E_0 = 33$ meV incident energy respectively.

We thus conclude that within experimental and simulation uncertainty, the current virtual experiment is in good agreement with Ref. [31].

4. Discussion and perspectives

In this paper, we have presented a new method for the simulation of complete neutron scattering experiments, including both instrument and sample. The propagation of neutrons through the instrument and the sample uses Monte Carlo ray-tracing. For interactions between neutrons and the sample, the main kinds of processes are accounted for, such as absorption, coherent and incoherent scattering, and multiple scattering. The structural and dynamic features of the sample are described by its dynamic structure factor, which is used to compute the energy and momentum transfers between neutrons and sample. Sample environment elements are treated in the same way as the sample. This method generates a simulated signal of a neutron scattering experiment and gives insight into the influence of the different contributions to the experimental signal, which are otherwise difficult to evaluate accurately. In a usual neutron scattering measurement, the empty-cell contribution should be subtracted from the total signal, taking into account the relative absorption ratio between the cell and the sample itself on the cell geometry. The virtual experiment provides simultaneously in a single computation all contributions in the detected signal, with their absolute intensities (counts per second) and cross-scattering dependency. Specifically, the usual empty-cell measurement is not required to extract the sample-only contribution. This is especially relevant for materials with large absorption cross section, like indium and mercury.

Results of simulated neutron scattering experiments are shown for liquid Rb and compared to previous measurements. Despite the fact that our virtual experiment is somewhat simpler than the real experiment, simulated results are in very good agreement with experimental analysis. The simulation model was entirely computed, and results only depend on the two-body potential, its core radius parameter, the instrument geometry and the measurement time.

In the example presented here, the single scattering, dynamic structure factor is given as input in the virtual experiment and multiple scattering in the total signal is then evaluated. In the case of real experiments, only the total scattering is known and programs that treat multiple scattering attempt to extract the single scattering contribution. A logical extension to the approach developed here will therefore be to implement a self-consistent scheme in which the total signal is given as input, the multiple scattering is evaluated and then subtracted from the total to give new input signal and the cycle is repeated until the final, simulated total signal matches the initial, measured total signal. However, as stated before, this process can in principle only be applied on half of the measured dynamical range, so that the final input signal will approximate the single scattering, dynamic structure factor.

While treating multiple scattering only requires a simple model instrument, the numerical approach embodied in the virtual experiment allows complex instrument configurations. Since the sample component can be used to describe any scatterer, the virtual experiment can include any scattering elements that are in the incident beam or are likely to be irradiated by the scattered beam. Practically this type of simulation can be used to identify spurious contributions to the total signal, which may come from a series of scattering events in several elements, for example sample (elastic scattering)-cryostat (elastic scattering)-sample (inelastic scattering). While cryostats are generally not so problematic, cryomagnets and pressure cells can involve significant amounts of scattering materials close to the sample position. Such a sample environment may be

difficult to improve upon, but our simulations would allow the ideal beam characteristics to be determined for a given set of scattering elements around the sample position.

Finally, we intend to make virtual experiments available to new and established users of neutron scattering facilities in order to allow them to understand how experiments work and to evaluate if a proposed experiment is feasible. When some knowledge about the container and/or the sample is available, it is possible to compute accurately their contributions in a measurement, providing potentially, invaluable help in data analysis.

Acknowledgments

We gratefully thank J.R.D. Copley, H. Fischer, D. Champion, A. Filhol and J. Ollivier for helpful discussions. We would like to thank Professor G. Kahl for providing us with the code for computing the liquid rubidium potential used in the classical molecular dynamics simulations. All simulations were performed using the computing resources available at the LL. *McStras* is supported as a JKA of the NMI3 network by the European Commission under the 6th Framework Programme through the Key Action: Strengthening the European Research Area, Research Infrastructures, Contract no.: RII3-CT-2003-505925.

References

- [1] H. Maier-Leibnitz, T. Springer, *Reactor Sci. Technol.* 17 (1963) 217.
- [2] D.F.R. Mildner, *Nucl. Instrum. Methods Phys. Res. A* 290 (1990) 189.
- [3] R.A. Lowe, *J. Nucl. Energy Part A: Reactor Sci.* 11 (1960) 69.
- [4] J.R.D. Copley, *Nucl. Instrum. Methods Phys. Res. A* 510 (2003) 318.
- [5] E. Fermi, J. Marshall, L. Marshall, *Phys. Rev.* 72 (1947) 193.
- [6] J. Peters, *Nucl. Instrum. Methods Phys. Res. A* 540 (2005) 419.
- [7] C.D. Clark, E.W.J. Mitchell, D.W. Palmer, I.H. Wilson, *J. Sci. Instrum.* 43 (1966) 1.
- [8] A.K. Freund, *Nucl. Instrum. Methods Phys. Res.* 213 (1983) 495.
- [9] V.E. Sears, *Acta Cryst.* A 53 (1997) 35.
- [10] G. Shirane, S.M. Shapiro, J.M. Tranquada, *Neutron Scattering with a Triple-Axis Spectrometer*, Cambridge University Press, 2002.
- [11] L. Alianelli, *J. Appl. Cryst.* 37 (2004) 732.
- [12] V. Radela, *IEEE Trans. Nucl. Sci.* NS-21 (1974) 51.
- [13] V. Peskov, G. Charpak, W. Dominik, F. Sauli, *Nucl. Instrum. Methods A* 277 (1989) 547.
- [14] G. Manzan, B. Guerard, F.A.F. Fraga, L.M.S. Margato, *Nucl. Instrum. Methods Phys. Res. A* 535 (2004) 102.
- [15] J.R.D. Copley, *J. Neutron Res.* 1 (1993) 21.
- [16] L.D. Cussen, *J. Appl. Cryst.* 36 (2003) 1204.
- [17] F. James, *Rep. Prog. Phys.* 43 (1980) 1145.
- [18] J.R.D. Copley, P. Verkerk, A.A. van Well, H. Fredrikze, *Comput. Phys. Commun.* 40 (1986) 337.
- [19] D.F.R. Mildner, C.A. Pellizari, J.M. Carpenter, *Acta Cryst.* A 33 (1977) 954.
- [20] P.A. Seeger, L.L. Daemen, T.G. Thelliez, R.P. Helm, *Phys. B* 283 (2000) 433. <<http://www.paseeger.com/>>.
- [21] J. Saroun, J. Kulda, *Phys. B* 234 (1997) 1102. <<http://omegaulf.cas.cz/restrax/>>.
- [22] P. Willendrup, E. Farhi, K. Lefmann, *Phys. B* 350 (2004) 735.
- [23] K. Lefmann, K. Nielsen, *Neutron News* 10 (1999) 20. <<http://www.mcsatas.org/>>.
- [24] W.-T. Lee, X.-L. Wang, *Neutron News* 13 (2002) 30.
- [25] E. Farhi, T. Hansen, A. Wildes, R. Ghosh, K. Lefmann, *Appl. Phys. A* 74 (2002) S1471.
- [26] C. Schanzer, P. Boni, U. Filges, T. Hils, *Nucl. Instrum. Methods Phys. Res. A* 529 (2004) 63.
- [27] G. Zsigmond, K. Leutenant, S. Manoshin, H.N. Bordallo, J.D.M. Champion, J. Peters, J.M. Carpenter, F. Mezei, *Nucl. Instrum. Methods Phys. Res. A* 529 (2004) 218.
- [28] K. Leutenant, *J. Phys.: Condens. Matter* 17 (2005) S167.
- [29] H. Schober, E. Farhi, F. Mezei, P. Allenspach, K. Andersen, P.M. Bentley, P. Christiansen, B. Cubitt, R.K. Heenan, J. Kulda, P. Langan, K. Lefmann, K. Leutenant, M. Monkenbusch, P. Willendrup, J. Saroun, P. Tindemans, G. Zsigmond, *Nucl. Instrum. Methods Phys. Res. A* 589 (2008) 34.
- [30] V. Hugouvieux, E. Farhi, M.R. Johnson, W. Kob, *Phys. B* 350 (2004) 151. See also V. Hugouvieux, A complete simulation of neutron scattering experiments, Ph.D. Manuscript, Univ. Montpellier, France, 2004.
- [31] J.R.D. Copley, J.M. Rowe, *Phys. Rev. A* 9 (1974) 1656.
- [32] V.E. Sears, *Adv. Phys.* 24 (1975) 1.
- [33] E.G. Bischoff, M.L. Yeater, W.E. Moore, *Nucl. Sci. Eng.* 48 (1972) 266.
- [34] M.W. Johnson, Harwell report, AERE – R 7682, March 1974.
- [35] J.R.D. Copley, *Comput. Phys. Commun.* 7 (1974) 289.
- [36] C. Petrillo, F. Sacchetti, *Acta Cryst.* A 46 (1990) 440.
- [37] G.L. Squires, *Thermal Neutron Scattering*, Cambridge University Press, 1978.
- [38] V.E. Sears, *Neutron News* 3 (1992) 26.
- [39] M. Matsumoto, T. Nishimura, *ACM Trans. Model. Comput. Simul.* 8 (1998) 3.
- [40] J.R.D. Copley, J.M. Rowe, *Phys. Rev. Lett.* 32 (1974) 49.
- [41] F. Demmel, A. Diebold, H. Aschauer, C. Morkel, *J. Non-Cryst. Sol.* 353 (2007) 3164.
- [42] J. Demmel, D. Pasqualini, C. Morkel, *Phys. Rev. B* 74 (2006) 184207.
- [43] J.-F. Wax, R. Albaladejo, J.-L. Brettonet, *Phys. Rev. B* 65 (2001) 014301.
- [44] R. Kleib, G.E. Ostrowski, D.L. Price, J.M. Rowe, *Nucl. Instrum. Methods* 106 (1973) 221.
- [45] P. Willendrup, U. Filges, L. Keller, E. Farhi, K. Lefmann, *Phys. B* 386 (2006) 1032.
- [46] P.H. Miller Jr., J.W.M. DuMond, *Phys. Rev.* 57 (1940) 198.
- [47] S. Kanbayashi, G. Kahl, *Phys. Rev. A* 46 (1992) 3255.
- [48] G. Kahl, S. Kanbayashi, *J. Phys.: Condens. Matter* 6 (1994) 10897.
- [49] T. Fog, K. Murzyn, K. Hinsen, G.R. Kneller, *J. Comput. Chem.* 24 (2003) 657.
- [50] W. Olsse, *Handbook of Thermodynamic and Transport Properties of Alkali Metals*, Ac. Press, Oxford, 1985.

iFit: A new data analysis framework. Applications for data reduction and optimization of neutron scattering instrument simulations with McStas

E. Farhi^{a,*}, Y. Debab^a and P. Willendrup^b

^aILL, Grenoble, France

^bTechnical University of Denmark, Kongens Lyngby, Denmark

Abstract. We present a new tool, iFit, which uses a single object class to hold any data set, and provides an extensive list of methods to import and export data, view, manipulate, apply mathematical operators, optimize problems and fit models to the data sets. Currently implemented using Matlab®, the toolbox is lightweight and comes with an extensive documentation based on tutorials with ready-to-run examples for each operator. Provided with the package is a set of optimization algorithms, which we have benchmarked in order to recommend the ones that provide the best success rate for both continuous and noisy problems. These optimizers can then be used to fit models onto data objects, and optimize McStas instrument simulations. As an application, we propose a methodology to analyse neutron scattering measurements in a pure Monte Carlo optimization procedure using McStas and iFit. As opposed to the conventional data reduction and analysis procedures, this new methodology is able to intrinsically account for most of the experimental effects, and results in the sample only model, de-convolved from the instrument.

Keywords: Numerical analysis, optimization, Monte Carlo, neutron scattering

1. Introduction

Data analysis software have a unique *raison d'être*: to shorten the path from a raw scientific result to publishable quality data. In order to accomplish this task, the software should be able to import, view, export, and analyse the data, including finding the best scientific model to interpret it. We may classify the data analysis software in two categories, those which may be used for all kinds of problems and data sets, and those that have been designed for specific tasks.

In the neutron scattering world, general scope packages are seldom seen. Most of them have been designed with a general kernel of functionalities, allowing more oriented data processing with plug-in or script-based extensions. The LAMP software [46] was probably the first attempt of this kind, followed by other more recent packages such as DAVE [1], DANSE [18] and Mantid [11] or even, in the spirit, the storage data format NeXus [33]. As an alternative, users may turn to commercial solutions.

However, many problem specific software have been proposed such as FullProf [47], CCSD [5], ICSD [2], GRASP [9], SASfit [34], nMOLDYN [7,32], Horace [45]. They all import data sets, and apply advanced data analysis, based upon reference publications with mathematical expressions in a sequence of corrections and analysis.

In this study, we present a generic software tool, iFit, which may handle any type of data, and instead of being single arrays, the symbols representing the data sets carry error bars, axes, monitors and other metadata. The

*Corresponding author: E. Farhi, ILL, BP 156, 38042 Grenoble Cedex 9, France. Tel.: +33 476 207 135; E-mail: farhi@ill.fr.

number of operators that can be applied on the data sets is large enough so that, in principle, any mathematical expression can be used, transforming the signal, error bars and axes along the process. Operators can cope transparently with differences in data sets binning and dimensionality appearing in expressions. In this context, this new package is just an evolution in the same direction, and perhaps slightly further away, as the other generic previous packages. The package in its current distribution is to be used only through commands at the prompt and scripts. No graphical user interface is provided, except e.g. the data visualization and optimization monitoring (see dedicated sections below). Compared with commercial packages, iFit can handle unlimited dimensionality data sets, seamlessly, both in mathematical operations and model fitting, and it accurately propagates the error on the signal for all operators.

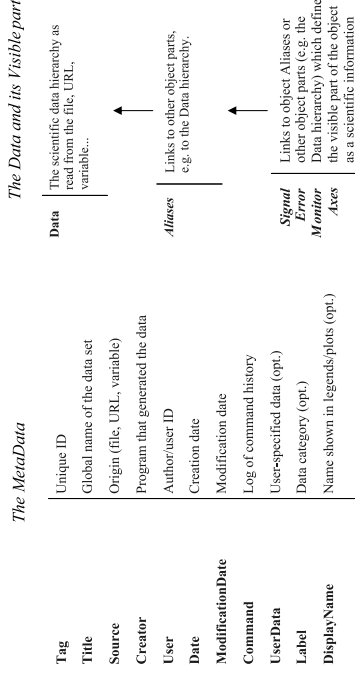
The iFit toolbox is freely available [25], with an extensive documentation based on a series of short tutorials for the different class of operators, including import and export of data files, plotting and visualization, fitting and optimization, and arithmetic operators. More specific data analysis, especially for neutron scattering data, can be carried out using these operators, as scripts or using graphical user interfaces to be written in the future by the developers and the neutron scatterers community. Accuracy and ease of use are among the most important criteria to judge the data analysis software, whereas portability and maintainability are key factors to ensure a long life-time and widely spread software. The limited size of the iFit project, the use of Matlab®, and the extensive documentation and tutorials enable to satisfy most of these requirements. The iFit concepts may be ported to Python in the future for better integration into e.g. Mantid [11].

2. Structure of data sets

In order to analyse scientific data, it is desirable to store it into a variable which can be processed by methods (e.g. mathematical operators). The storage area should allow to access all parts from the data, but also give complete freedom in the way to assign the data in terms of scientific signal, related error bars and monitors (e.g. counting time or incident flux record), as well as axes (time, angle, height, variable scan, ...). This interpretation defines the visible part of the data set. But such a definition may possibly be changed in the course of the data analysis, e.g. explore an other signal stored in the same data set, use the same signal with different axes definitions, perform coordinate changes and apply the corresponding Jacobian, etc. This concept goes beyond a single array stored in a variable, as it carries information about how to display the data, in a way similar to the NeXus format definition [33].

In the iFit implementation, all data sets are stored into objects, using a single class definition named *iData*. A schematic view of the iData object structure is shown in Fig. 1. These objects are essentially composed of four distinct parts:

1. The *MetaData* provides information to identify the object and relate it to the origin of the data (author, date, title, file, ...). A *Tag* (unique ID) is assigned to each object, but the user may define a *Label* (to store a data set category, such as 'Empty cell', 'Background', 'Test Run', ...) and a *DisplayName* which associates a specific name to the object.
2. The scientific data itself, e.g. read from a file, is stored as a structured hierarchy in the *Data* field of the object.
3. Once the object has been created, the data can be accessed directly from the *Data* field above. However, in order to ease the access to the data, it is possible to define short-cuts in each object, named *Aliases*. An alias is a name which links to e.g. a portion of the data hierarchy, or to other aliases. Usually, one alias may be defined for each significant scientific information such as detector counts, temperature records, etc. in the same object. Such aliases enable an easy access to various parts of the data, whatever be its internal hierarchy.
4. Finally, in order to plot the data and perform mathematical operations, the visible part of the object is defined as mandatory aliases in terms of a *Signal*, as well as *Error* bars on this signal, and an associated *Monitor* (e.g. counting time). When not assigned, the signal is set as an alias to the largest numerical block (vector, matrix, ...) in the data hierarchy. The error is then set to the square root of the signal and the monitor is set to 1. In order to represent a scientific data set, *Axes* should as well be defined as aliases. Undefined axes are set as the row, columns, pages, etc. indices in the signal. In this implementation, no error bar is handled along the axes.

Fig. 1. Schematic structure of the *iData* object structure handled in iFit.

Most of the object internal definitions are set automatically upon reading of the data files. However, it is always possible to either directly modify the data hierarchy, and redefine some of the aliases, as well as the signal, error, monitor and axes. Dedicated object methods have been defined to get and set all parts of the object, including aliases and axes.

Any data set dimensionality can be handled (including with complex numbers data), as well as arrays of objects for vectorial/iterative processing of many data sets.

3. Importation of data files, exportation of results

As the iFit package is designed as a generic data analysis tool, to hold any kind of scientific data, it is desirable to be able to read a large variety of file formats and data sources. A list of supported formats is shown in Table 1. All importation methods are lossless (the whole file content is retained) and transparent (do not require to tune the import process). In particular, the importation methods in iFit allow to read *any* text file, whatever be its internal content and comments. Distant and compressed files are also handled. The syntax to load the data into an *iData* object is to convert its file name to the object class:

```
>> a = iData('filename');
```

which creates an object *a* containing the data file. Lists of file names and directory paths produce an array of objects. In some cases, the signal, error, monitor and axes may require to be re-defined after the importation.

Most of the supported data formats can also be used to write back the object contents into files. However, some of the exportation formats are not lossless (some information and initial data structure may be lost). The syntax to save an object *a* to a file is:

```
>> save(a, 'filename.ext');
```

where the file extension defines the format.

4. Viewing data sets

Once a data set has been stored into an object, it is possible to display its visible part (signal/monitor, axes) simply with the command:

Table 1
Data formats supported by iFit (importation/exportation)

Format (load)	Read	Write
Any free format text, including ILL data, SPEC (ESRF), McStas data files, INX, ISIS/SPE, CSV (comma separated values), Chalk River CNBC files	Yes	Yes
HDF 4, HDF 5, NeXus and all derivatives, NetCDF	Yes	Yes
Matlab Mat-file (MAT), figure (FIG) and variables	Yes	Yes
Excel spreadsheet (XLS)	Yes	Yes
GIF BMP PNG TIFF JPEG ICO Images, FITS astronomical image	Yes	Yes
ESRF Data format (EDF), Analyze MRI data format (HDR/IMG),	Yes	Yes
Protein Data Bank (PDB), Crystallographic Binary File (CBF/imgCIF), XML description file, Lotus 1-2-3, NeXT/SUN (.au)	Yes	Yes
sound, Microsoft WAVE sound (WAV), Audio/Video interleaved multimedia container (AVI), MarCCD, Platus X-ray cameras	Yes	
Distant URL (ftp:// https://) and compressed files (ZIP GZip, TAR, Z)	Yes	
PostScript (PS, EPS), Adobe PDF and Illustrator, Virtual Reality world (VRML), Scalable Vector Graphics (SVG)	Yes	Yes

>> plot(a);

which can handle vectorial, matrices (surfaces) and 3D tensor (volumes) data sets, as shown in Fig. 2. Other representations are also possible, such as waterfall (side-by-side), superposed (stacked), RGB images, contour plots, etc. The graphics produce a camera-ready rendering quality, with graphical tools to export, edit, pan rotate, and explore the view.

The data set information (internal definitions, metadata, statistics, ...) can be displayed with:

>> disp(a).

5. Operators

The iFit toolbox contains currently 172 operators which can be used to manipulate data sets, among which 80 are mathematical operators, 64 are associated to data manipulation and 28 relate to plotting. Each operator takes care to propagate error bars along the data analysis and continuously checks the objects consistency. All methods have been written to be as simple as possible to use, and stick whenever appropriate to the usual simple matrix operators defined in Matlab®. All of the iFit methods can be used with single and arrays of objects for iterative processing.

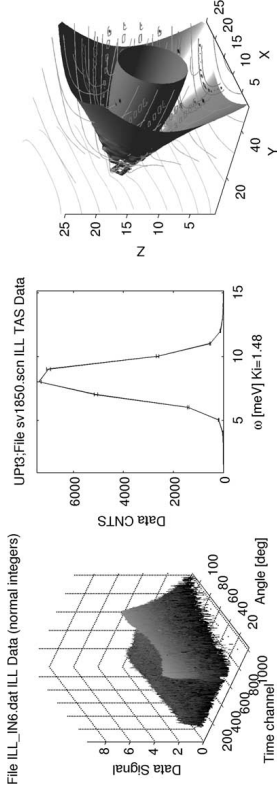
5.1. Manipulating data sets (cutting, appending)

To extract a portion of an object, a mere sub-indexing is needed, such as:

>> b = a(5:10,:);

where we use the ':' symbol to denote a range, which used alone refer to the whole available range in the data. The results is an identical or lower dimensionality object. Similarly, it is possible to concatenate many objects, or create a higher dimensionality object, such as for instance many 1D objects assembled into a 2D surface with side-by-side appending along the rank 2 dimension:

>> b = cat(2,[array of 1D objects]).

Fig. 2. Graphical rendering of 1D, 2D and 3D *Data* objects.Table 2
Mathematical unary operators (selection) applied on an object (signal, error)

Operator	New signal	New error	Description
acos	$\cos^{-1}(\text{signal})$	$-\text{error} / \sqrt{(1 - \text{signal}^2)}$	Arc cosine. Signal should be $[-1 : 1]$
asin	$\sin^{-1}(\text{signal})$	$\text{error} / \sqrt{(1 - \text{signal}^2)}$	Arc sine. Signal should be $[-1 : 1]$
atan	$\tan^{-1}(\text{signal})$	$\text{error} / (1 + \text{signal}^2)$	Arc tangent.
cos	$\cos(\text{signal})$	$-\text{error} \cdot \sin(\text{signal})$	Cosine. Signal is in [rad]
sin	$\sin(\text{signal})$	$\text{error} \cdot \cos(\text{signal})$	Sine. Signal is in [rad]
tan	$\tan(\text{signal})$	$\text{error} / \cos^2(\text{signal})$	Tangent. Signal is in [rad]
exp	e^{signal}	$\text{error} \cdot e^{\text{signal}}$	Exponential
log	$\ln(\text{signal})$	$\text{error} / \text{signal}$	Logarithm (Neperian, base 2)
log 10	$\log_{10}(\text{signal})$	$\text{error} / (\ln(10) \cdot \text{signal})$	Logarithm (base 10)
sqrt	$\sqrt{\text{signal}}$	$\text{error} / (2 \cdot \sqrt{\text{signal}})$	Square root

5.2. Unary and binary mathematical operators

A selection of the unary operators is listed in Table 2. The binary operators use the monitor value as a weight for the operands. A selection of such binary operators is shown in Table 3. In case the axes ranges and binning of the operands do not match, a re-binning of the axes is performed transparently. Additionally, when one of the two objects has a lower dimensionality, it is extended along the missing dimensions (by simple duplication) to match the other higher dimensionality, so that the operation can be performed. The combine operator enables to merge data sets from separate measurements in order to improve the overall statistical accuracy, as it corresponds to a weighted addition.

5.3. Advanced mathematical operators

These operators include an extended set of mathematical operations, as well as specific methods to cut, slice, append, project, integrate, derive, and interpolate/re-bin data sets.

In addition to the previous unary and binary operators, iFit provides methods to compute the Fourier transform (FFT) of objects (with axes being converted to their reciprocal space), the derivatives (gradient) as well as the Laplacian, the integrals along axes, the correlation and the convolution of two objects e.g. to handle resolution function effects (which require one of the two objects to be normalized and centred), the projections of an object onto one of the axes, and the interpolation of one object onto another axes system. In the case of axes coordinate changes, the Jacobian of the transformation can also be computed automatically.

Table 3
Mathematical binary operators (selection)

Operator	Signal ($s3$)	Error ($e3$)	Monitor ($m3$)	Description
plus <i>and</i> +	$m3 \cdot (s1/m1 + s2/m2)$	$\sqrt{(e1/m1)^2 + (e2/m2)^2}$	$m1 + m2$	addition
minus <i>and</i> -	$m3 \cdot (s1/m1 - s2/m2)$	$\sqrt{(e1/m1)^2 + (e2/m2)^2}$	$m1 + m2$	subtraction
times <i>and</i> *	$s1 \cdot s2$	$s3 \cdot [(e1/s1)^2 + (e2/s2)^2]$	$m1 \cdot m2$	multiplication (<i>element by element</i>)
minuses <i>and</i> *	$s1 \times s2$	$e1 \times e2$	$m1 \times m2$	matrix multiplication
divide <i>and</i> /	$m2^2 \cdot (s1/s2)$	$s3 \cdot [(e1/s1)^2 + (e2/s2)^2]$	$m1 \cdot m2$	division (<i>element by element</i>)
combine <i>and</i> \	$s1 + s2$	$\sqrt{(e1^2 + e2^2)}$	$m1 + m2$	add signals and monitors independently
power <i>and</i> ^	$m3 \cdot (s1/m1)^{(s2/m2)}$	$s3 \cdot [s2 \cdot e1/s1 + e2 \cdot \ln(s1)]$	$m1^{m2}$	exponentiation
comparisons:	comparison($s1, s2$)	$\sqrt{(e1/m1)^2 + (e2/m2)^2} / [(s1/m1 + s2/m2)]$	1	binary comparisons

lt <, le <=,
gt >, ge >=,
ne ~, eq ==

Notes: The two initial objects ($s1, e1, m1$) and ($s2, e2, m2$) produce the object ($s3, e3, m3$), where s, e and m refer to the signal, the error on the signal and the monitor associated to the measurement, respectively.

Some operators have also been written to obtain simple statistical information about the objects (minimum, maximum, mean, variance) as well as an automatic peak searching capability along any of the defined axes.

6. Optimization methods for fitting models

6.1. Benchmarking optimization methods

Optimization methods may be used in many engineering, economics, mathematical, biology, chemistry and physics problems. This includes curve fitting, as well as finding any set of parameters to maximize or minimize a given criteria, also known as cost or objective function. There exists a wide set of solvers, most of them being iterative, based upon different mathematical approaches [3,22,35,43,44]. Hessian and gradient based methods are known to be fast for a limited number of parameters, and their convergence is guaranteed, that is require a finite number of iteration steps to identify an optimal solution. But the derivatives estimate is computationally intensive for large dimensionality problems, and requires the cost function to be continuous differentiable (and in particular noiseless). This class of solvers includes the steepest-descent Newton-like methods such as the Levenberg-Marquardt one. As the search is driven by derivatives, that is a local description of the cost function in a Taylor's series, such methods can easily miss global solutions, and be trapped around local solutions. Derivative-free deterministic methods, such as the well known simplex, are usually less sensitive to noise in the cost function, but are often slower. Last, the heuristic methods, such as the genetic algorithm, simulated annealing, and particle swarm optimizers, do not ensure convergence during the search, and only provide approximate solutions. However, as most of these heuristic make use of quasi-random numbers in the search algorithm, they are able to thrive in noisy problems where gradient methods fail.

Within iFit, we have carried out a cross-comparison of a wide set of optimization algorithms, developed by various contributors, to estimate a mean optimization success rate in solving a problem. Of course, the success rate of a given optimization method depends drastically on the problem to solve, and the starting parameter set. In particular, some optimizers have been designed specifically to solve a reduced set of problems. However, we have here equally tested all optimizers on a common set of functions to minimize. Even though this test suite does not ensure that an optimizer will solve a peculiar problem not part of the suite, it provides a common work bench for

comparing all these methods. Compared to previous optimizers benchmarking work, we have gathered a larger set of optimization algorithm. Also, we have tested these optimizers both on continuous problems, but also on noisy problems, which are common in neutron scattering measurements and computations.

In this study, we have compared the convergence success rate and speed of 21 different optimization methods, to minimize 52 different multi-dimensional objective functions [20,39,54] which global minimum is known, corresponding with a zero cost function value. The objective function test suite includes functions with a flat minimum region, and many local minima solutions aside a global solution, as well as function with discontinuities and minimum on the borders of the search area. The choice of optimizers, as shown in Table 4, has been determined by their availability to non-experts [10,17,26,28,31,40,49], and wrapped to match the default Matlab optimizer *fminsearch* syntax. For each objective function F to minimize (within 52), we cast randomly an initial parameter set R in a search volume of dimensionality N (within 1, 2, 4, 6, 8, 10, 12, 16, 20, 24, 28, 32, 40, 48 and 64).

Table 4
Optimizers available with iFit, obtained from [10,17,26,28,31,40,49]

Function name	Description	Continuous problems		Noisy problems	
		Success rate [%]	Function calls	Success rate [%]	Function calls
fminanneal	Simulated annealing [31]	83	407	21	786
fminbfgs	Broyden–Fletcher–Goldfarb–Shanno [6,16,19,50]	76	114	3	53
fmincgtrust	Steihaug Newton-CG-Trust [6,16,19,28,50]	88	248	14	946
fmincmaes	Evolution strategy with covariance matrix adaptation [20,21]	89	1878	71	2066
fminga	Genetic algorithm (real coding)	97	5451	78	5677
fmingradrand	Random gradient [51]	78	1264	26	941
fminhooke	Hooke–Jeeves direct search [23,27,28]	97	1176	56	3067
fminimfil	Implicit filtering [28]	93	1424	54	2805
fminkalman	Unscented Kalman filter [26]	76	1687	37	4173
fminlm	Levenberg–Marquardt [16,17]	80	1033	5	6209
fminnewton	Newton gradient search [28]	77	701	21	2138
fminpowell	Powell search [4]	99	324	52	4484
fminpso	Particle swarm optimization [10,29]	98	2758	84	3080
fminralg	Shor R-algorithm [53]	89	166	16	586
fminrand	adaptive random search [49]	75	4240	62	4318
fminscse	Shuffled complex evolution [10,12]	96	3233	85	3185
fminsearchbnd	Nelder–Mead simplex (<i>fminsearch</i>) [36]	66	740	11	2028
fminsimplex	Nelder–Mead simplex (alternate implementation than <i>fminsearch</i>) [42]	82	465	40	525
fminsimpsa	Simplex/simulated annealing [8,10,30]	97	2795	85	3672
fminswarm	Particle swarm optimizer (alternate implementation than <i>fminpso</i>) [52]	95	2512	75	3085
fminswarmhybrid	Hybrid particle swarm optimizer [29,52]	92	1671	47	4032

Notes: The four last columns indicate the mean success percentage rate (higher is better) and function calls (cost, lower is better) per optimizer obtained with the benchmarking procedure, for both continuous and noisy cost functions. Repetitions of the benchmarking procedure enable to determine the uncertainty on the mean success percentage rate and the number of function calls as about 5%.

All optimizers O , within 21 listed in Table 4, are then executed, using their default configurations and the same stopping conditions for each test. This procedure is repeated $M = 50$ times with different random starting points R , in order to minimize systematic errors that may arise from the selection of the initial starting point. The same benchmarking procedure has been repeated when adding a 10% Gaussian noise to the cost function. The optimizer is counted as successful when the final cost function value is within the function tolerance δF , which is set to 10^{-4} and 10^{-3} respectively for continuous and noisy objective functions. This defines a convergence volume around the global minimum. The convergence criteria on the parameter change δX is not used, and set to 0. Once all functions have been minimized with all optimization routines and all starting points, we sum-up the number of optimization success events along objective functions F per optimizer O – which provide the optimizer mean success rate. The maximum number of function evaluations (budget) is set to $2500 * N$. The maximum number of algorithm iterations during the optimization is set to $250 * N$. The time required to find successful solutions is proportional to the number of objective function iterations. The other optimizer settings are detailed in Table 5, are kept to their default value as defined from the original source code [10,17,26,28,31,40,49].

The detailed results of the benchmarking procedure enable to recommend high-success and fast optimizers, for both continuous and noisy objective functions [25], as shown in Fig. 3. The corresponding mean success rate (that is the mean probability to solve a problem) and number of cost function iterations (proportional to the solving time required) for all dimensionality values is shown in Fig. 3 and Table 4. Some of the algorithms which are provided in more than one implementation (see for instance the swarm and simplex methods) behave differently both in success rate and speed. This demonstrates the importance of the coding quality.

In short, gradient based optimizers perform better than most other methods when the objective function is smooth and not noisy. We may then recommend the following optimizers for continuous problems:

- *The fastest methods* are the Shor R-algorithm and the Powell methods.
- *The most reliable methods* are the Particle Swarm Optimization, the Powell, the Hooke and the Simplex/Simulated Annealing methods.

Alternatively, optimizers that make use of random search, such as swarms, are better suited to solve noisy objective functions. We may then recommend the following optimizers for noisy problems:

- *The fastest methods* are the Evolution Strategy with Covariance Matrix Adaptation and the Particle Swarm Optimization.
- *The most reliable methods* are the Swarm and the Simplex/Simulated Annealing.
- *For large dimensionality* (many parameters), the Evolution Strategy with Covariance Matrix Adaptation as well as the Shuffled Complex Evolution.

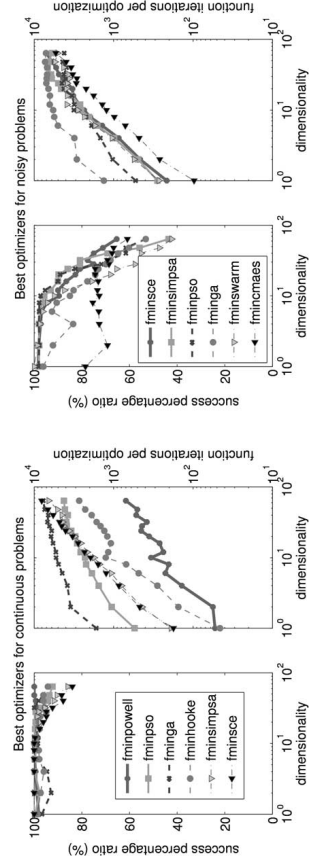


Fig. 3. Mean success ratio and number of cost function iterations of some of the best optimizers benchmarked in this study, for continuous (left) and noisy problems (right). The success ratio and number of cost function iterations is computed for the part of the test suite which could be solved for each optimizer and dimensionality. The name of the optimizers refers to the Tables 4 and 5.

Table 5

Default settings used in optimizers available in iFit		
Function name	Description	Default settings
fminanneal	Simulated annealing [31]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 100 per temperature, MaxFunEvals = 5000, $T_0 = 1$, $T_{\text{stop}} = 10^{-8}$, $T_{n+1} = 0.8 * T_n$, generator = Gaussian noise on one parameter
fminbfgs	Broyden-Fletcher-Goldfarb-Shanno [6,16,19,50]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = $20 * N$, MaxFunEvals = 5000
fmincgtrust	Stihaug Newton-CG-Trust [6,16,19,28,50]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = $100 * N$, MaxFunEvals = 10,000
fmincmaes	Evolution Strategy with Covariance Matrix Adaptation [20,21]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = ∞ , number of agents = $4 + 3 * \ln(N)$
fminnga	Genetic Algorithm (real coding)	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 10,000, number of agents = 50, number of mutations = 40
fmingradrand	Random Gradient [51]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 1000, 5% Gaussian noise
fminhooke	Hooke-Jeeves direct search [23,27,28]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = $\min(20, 10 * N)$, MaxFunEvals = 1000, scales = $1, \frac{1}{2}, \dots, \frac{1}{256}$
fminimfil	Implicit filtering [28]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 1000, with BFGS quasi-Newton local search
fminkalman	Unscented Kalman filter [26]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 1000, $P = X $, $Q = 10^{-6}$
fminlm	Levenberg-Marquardt [16,17]	$\delta F = 10^{-5}$, $\delta X = 10^{-4}$, MaxIter = 5000, MaxFunEvals = 10,000
fminnewton	Newton gradient search [28]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = $10 * N$, MaxFunEvals = 10,000
fminpowell	Powell search [4]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, with Coggins local line search
fminpsp	Particle swarm optimization [10,29]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, $C_1 = 2.8$, $C_2 = 1.3$, number of agents = 25
fminralg	Shor R-algorithm [53]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, space dilatation factor = 2.5
fminrand	Adaptive random search [49]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, number of agents = $3 * N$, contraction factor = 0.2,
fminsee	Shuffled complex evolution [10,12]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, number of agents = $5 * (2 * N + 1)$
fminsearchbnd	Nelder-Mead simplex (<i>fminsearch</i>) [36]	$\delta F = 10^{-4}$, $\delta X = 10^{-4}$, MaxIter = $200 * N$, MaxFunEvals = $200 * N$
fminsimplex	Nelder-Mead simplex (alternate implementation than <i>fminsearch</i>) [42]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000
fminsimpsa	Simplex/simulated annealing [8,10,30]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, number of agents = 50, cooling factor = 0.9, cooling rate = 10
fminswarm	Particle swarm optimizer (alternate implementation than <i>fminpsp</i>) [52]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, $C_1 = 2$, $C_2 = 2$, $W = 0$, number of agents = 20
fminswarmhybrid	Hybrid particle swarm optimizer [29,52]	$\delta F = 10^{-3}$, $\delta X = 10^{-8}$, MaxIter = 1000, MaxFunEvals = 5000, $C_1 = 2$, $C_2 = 2$, $W = 0$, number of agents = 20, with local simplex search

Notes: δF and δX are the convergence criteria for the final cost function value and variation, and the optimization parameter change respectively. N is the problem dimensionality (number of parameters to optimize). *MaxIter* and *MaxFunEvals* are the maximum number of algorithm iterations and the maximum number of cost function evaluations respectively before aborting the optimization. These default settings are determined from the original contributed code [10,17,26,28,31,40,49].

6.2. Minimizing an objective function

Given an objective function c of parameters p , the syntax to find the minimum of c starting from an initial parameter set p_0 is:

```
>> p = optimizer(c, p0);
```

where the optimizer is one of the methods listed in Table 4. Optional arguments enable to change the optimizer algorithm configuration, as detailed in Table 5, as well as to define some constraints on the parameters during the search (e.g. minimum and maximum values, fixed parameters, maximal variation of parameters between two iterations, ...). The optimization procedure can return additional results, including the uncertainty on the optimal parameter set. It is also possible to visually monitor the optimization procedure by a dynamic display of the objective function and parameters evolution.

6.3. Fitting a parametrized model onto a data set

Once the data sets can be loaded, and optimizers are available, it is straightforward to implement a fitting procedure. In this case, the principle is to find the best parameter set of a model function, which match the data set. The criteria for the fit is usually the least-square one $c = \chi^2$:

$$\chi^2 = \sum (\text{signal}/\text{monitor} - \text{model})^2 / \text{error}^2,$$

where the signal, error and monitor values are obtained from the data set. Other criteria definitions are available. The model value is obtained as:

$$\text{model} = f(p, \text{axes}),$$

where f is the model function of parameters p using axes from the data set. A set of usual model function (Gaussian, Lorentzian, Voigt, ...) is provided with iFit, as well as a tool to create automatically such models, by giving their mathematical expression. Multidimensional fits are supported, even with lower dimensionality model functions, which are then extended to match the data set. For instance fitting a 2D data set with a simple 1D Gaussian model *de-facto* extends the model to the product of two Gaussian functions on each data set axis.

The general syntax for fitting a model f to the data set is:

```
>> p = fits(a, f);
```

which guesses a reasonable starting parameter set. Additional arguments allow to specify a starting parameter set for the search, as well as the optimizer to use (in Table 4) and its internal settings (see Table 5), constraints on parameters during the optimization, the criteria definition to use (for other choices than χ^2), ...

6.4. Optimization of neutron scattering instrument simulations with McStas

It is possible to define the criteria for the optimization as the result of a McStas ray-tracing neutron scattering instrument simulation [37,41,56], following the Section 6.2 and taking into account that iFit can transparently import the files generated by the simulation. In this case, we aim at maximizing some or all of the monitors and detectors located in the model, with the syntax:

```
>> p = mcstas(instrument, parameters, options);
```

where *instrument* is the name of the instrument description file for McStas, *parameters* is a named structure which lists the parameters used to initiate the search, and *options* allow to specify both some additional instrument simulation arguments (number of neutron events, number of processors, monitor names to use as criteria, ...) the figure of merit to use, as well as the optimizer configuration (name within Table 4, internal settings, constraints, ...).

7. Data reduction for neutron scattering experiments: Analytical corrections and use of virtual experiments with McStas

The usual method to correct data from neutron scattering measurements is to follow a series of subtractions (e.g. empty sample cell), normalizations (e.g. detector efficiency, reference sample, transmission), grouping, and variable change, etc. leading to a physically meaningful data set, on which models can further be tested. The corresponding model parameters, which are the relevant information for the scientist, are then often found by a fitting method, including instrument resolution effects. All of these operations are possible within the existing *iFit* toolbox, and can be decomposed into a series of simple mathematical operations. Once the data sets have been imported into memory, the whole *iFit* infrastructure ensures a transparent and simple use of each operator, seamlessly transforming the signal, error bars, monitor and axes.

A more advanced data reduction path resides in the use of virtual experiments [14,24,38]. If we define the virtual experiment model to produce the same type of data as the real measurement (same axes and dimensionality), then the simulated data may be compared with the real experiment and it is possible to define e.g. a least-square criteria between the real and simulated measurements. This way the model function used for the fit is the virtual experiment itself, and the parameters describe the instrument and sample configuration which should be as complete and accurate as possible.

Compared with an analytical method, the virtual experiment data reduction methodology has a number of advantages, but also a few drawbacks. From a real instrument, one may either model the resolution function, or measure it using reference samples (e.g. water for small angle instruments, NAC powder for powder diffractometers, Vanadium for time-of-flight and triple axis spectrometers, ...). In both cases, the obtained information is usually a projection of the resolution function onto the final detector signal, or an analytical model. Analytical resolution function models also assume a specific instrument geometry and configuration parameters, which are simplified compared with a full instrument description as handled by neutron ray-tracing Monte Carlo codes such as McStas, *Vitecs* [55] and *Restrax* [48]. On the other hand, the resolution function included in a virtual experiment implicitly contains all the correlations between the neutron state parameters (position, divergence, wavelength, spin, detection time, ...). The projection of this simulated resolution function on the detected signal proves in practice to be very close to the actual measurements, as reported in [13,57]. The virtual experiment model should be as close as possible to the real instrument geometry, which requires the scientist to know in depth the complete geometry and physical parameters of the elements composing the instrument. In practice, all instruments evolve with time, and it is rare that the real instrument perfectly corresponds with its blueprints and design specifications. This remark also holds for analytic instrument models, which parameters often have to be tuned in order to match the actual measurements. In this case, a discrepancy in the simulated instrument characteristics, compared with measurements, may reveal inconsistencies which in the end help in identifying parasitic effects. Such effects would often be taken as is when using the raw measured instrument resolution, without necessarily searching for their origin. Last, each virtual experiment may require substantially more computation time than a simple convolution with a measured or analytical model. But in practice, a virtual instrument model may be evaluated within a few seconds on modern computers with multi-core architectures. An optimization procedure using the virtual experiment as the model function, from which sample parameters are to be extracted, can then be carried out within minutes, depending on the optimizer choice and the number of parameters to refine. The particle swarm optimization, even though relatively slow, will provide an excellent success rate. We also point out that the multi-dimensional convolution between the sample scattering and the instrument resolution, is intrinsically included in the virtual experiment, whereas this operation must be specifically performed in a classical data analysis procedure.

McStas [37,41,56] provides an extended set of advanced components to model all parts of the neutron scattering instruments [14]. Not only neutron optics are available, but also complex material models to be used for the sample and e.g. all materials in the beam. These models usually take into account coherent and incoherent neutron scattering [57], self shielding (secondary extinction) and for some of them more advanced processes such as multiple neutron scattering [13]. Gas detectors can also be modelled in a very advanced description taking into account charge creation and drift before detection on multi-wires [14,15]. The McStas model may as well handle some

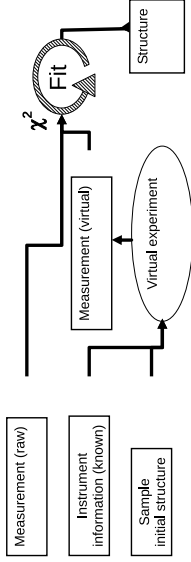


Fig. 4. Schematic diagram showing the fit procedure to compare raw data neutron scattering measurement with the corresponding virtual experiment model, in the case of a structure refinement. Upon optimization, the final solution provides sample parameters which best match the measurement, keeping the instrument configuration constant.

imperfections – such as, for instance, mirror damage, neutron optics misalignments and choppers drift – from the instrument design specifications. To date, most available McStas components, including sample kernels, have been validated with cross-comparison studies and measurements.

Using iFit, the only requirement is to define a model function which takes as input the instrument parameters e.g. incoming neutron wavelength, distances and geometry of the instrument components, neutron optics materials, . . . usually kept fixed during the fit procedure. These parameters may possibly be left slightly adjustable in order to cope with the real instrument imperfections. The sample model parameters should also be part of the virtual experiment, and are left free, within reasonable limits. Typical sample parameters may be a lattice structure with e.g. a reflection list or a parametrized structure factor $S(q)$, or a dynamical structure factor $S(q, w)$. The model then launches the virtual experiment and finally reads the simulated detector file, which is compared to the actual measurement. The procedure is iterated in an optimization loop, as shown in Fig. 4 in the case of a structure refinement. The result of this complex data analysis procedure are the refined sample parameters, fully corrected and decoupled from instrumental effects. The limitation of the virtual experiment resides in the completeness of the physics included in the simulated components and sample used in the description of the experiment, as well as the knowledge of the instrument by the scientist. But similar, limitations also exist in the analytical approach.

We estimate that the above mentioned procedure may potentially improve the data analysis of measurements, by providing a more accurate estimate of sample parameters. Achievements of this technique in the case of a structure refinement will be published in a separate paper, and compared with the analytical method results.

Acknowledgements

The authors are grateful to A. Filhol, Y. Raoul, L. Udby for their continuous encouragements and advices. P. Bentley convinced us to use swarm type optimizers, and this indeed proved to be an excellent suggestion. J. Ollivier pushed us to import time-of-flight data fast, which resulted in dramatic efficiency improvements. C. Rivasseau supported EF when morale was low. H. Jacobsen thoroughly tested the software, which greatly helped in providing a stable package. The authors also acknowledge the ILL for providing the resources to develop this software, both in time and equipment. We are grateful to authors Eric Ludlam, Felix Morsdorf, Joe Conti, Douglas M. Schwarz, Alexandros Leontitis, F. Sigworth, Argimiro R. Secchi, Javad Ivakpour, Nikolaus Hansen, Alexei Kuntsevich and Franz Kappel, C.T. Kelley, Brecht Donckels, Miroslav Balda, Paul Spencer, Juerg Schwizer, Daniel Buckton, Petr Mikulík, David Gingras, Joachim Vandekerckhove, Yi Cao, Sachin A. Nikumbh, Oliver Bunk, R.G. Abraham and J. Ollivier who have made available their Matlab open-source code, which has been included in the package.

References

- [1] R.T. Azuah, L.R. Kneller, Y. Qiu, P.L. W. Tregenna-Piggott, C.M. Brown, J.R.D. Copley and R.M. Dimeo, *J. Res. Natl. Inst. Stan. Technol.* **114** (2009), 341; see also <http://www.nist.gov/dave/index.html>.
- [2] A. Belsky, M. Hellenbrand, V.L. Karen and P. Luksch, *Acta Cryst.* **B58** (2002), 364–369; see also H. Hewat and P. Hewat, ILL, France, available at: <http://icsd.ill.fr>.
- [3] M.J. Box, *Comput. J.* **8** (1965), 42.
- [4] Brent, *Algorithms for Minimization Without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [5] J.P. Brown, *J. Neut. Res.* **4** (1996), 25; see also *Cambridge Crystallographic Subroutine Library*, available at: https://www.ill.eu/other_sites/cs/ftml/csdoc.html.
- [6] C.G. Broyden, *J. Inst. Math. Appl.* **6** (1970), 76–90.
- [7] V. Calandrimi, E. Pellegrini, P. Calligaris, K. Hinsien and G.R. Kneller, *Collection SFN* **12** (2011), 201–232.
- [8] Cardoso, Saleedo, Fejo de Azevedo and Bardosa, *Comp. Chem Eng.* **21** (1997), 1349.
- [9] C. Dewhurst, GRASP-ILL, France, available at: <http://www.ill.eu/instruments-support/instruments-groups/lss/grasp/home/>.
- [10] B. Donckels, BIOMATH, 2006, available at: <http://biomath.ugent.be/~brecht/downloads.html>.
- [11] N. Draper, M. Gyge, R. Taylor, M. Doucet, S. Campbell et al., Mantid, available at: <http://mantidproject.org>.
- [12] Q.Y. Duan et al., *J. Opt. Th. Appl.* **76** (1993), 501.
- [13] E. Farhi, V. Hugouvieux, M.R. Johnson and W. Kob, *Journal of Computational Physics* **228** (2009), 5251–5261.
- [14] E. Farhi and P. Willendrup, *Collection SFN* **12** (2011), 303–339.
- [15] E. Farhi and P. Willendrup, Advanced sources and optical components for the McStas neutron scattering instrument simulation package, *Collection SFN* (2012).
- [16] R. Fletcher, *Computer J.* **13** (1970), 317–322.
- [17] R. Fletcher, Report AERE-R 6799, Harwell, 1971; see also <http://www.mathworks.com/matlabcentral/fileexchange/17534>.
- [18] B. Fulz et al., DANSE, Caltech USA, available at: <http://danse.us>.
- [19] D. Goldfarb, *Mathematics of Computation* **24** (1970), 23–26.
- [20] N. Hansen and S. Kern, Evaluating the CMA evolution strategy on multimodal test functions, in: *Eighth Int. Conf. on Parallel Problem Solving from Nature PPSN VIII*, Lecture Notes in Computer Science, Vol. 3242, Springer, 2004, pp. 282–291.
- [21] N. Hansen and A. Ostermeier, *Evolutionary Computation* **9** (2001), 159–195; see also <http://www.lri.fr/~hansen/cmaesintro.html>.
- [22] J.T. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, 1992.
- [23] R. Hooge and T.A. Reeves, *Journal of the ACM* **8** (1961), 212.
- [24] V. Hugouvieux, E. Farhi and M.R. Johnson, *Physica B* **350** (2004), 151.
- [25] IFit web site <http://fit.mcode.org>.
- [26] R.E. Kalman, *Transactions of the ASME – J. Basic Engineering* **82** (1960), 35; see also <http://www.mathworks.com/matlabcentral/fileexchange/18217>.
- [27] A.F. Kaup Jr., *Communications of the ACM* **6** (1963), 313.
- [28] C.T. Kelley, Iterative methods for optimization, in: *Frontiers in Applied Mathematics*, Vol. 18, SIAM, Philadelphia, PA, 1999; see also <http://www.siam.org/books/kelley/fit18/index.php>.
- [29] J. Kennedy and R.C. Eberhart, Particle swarm optimization, in: *Proc. IEEE Conf. on Neural Networks, Vol. IV, Piscataway, NJ*, 1995, pp. 1942–1948.
- [30] Kirkpatrick, *J. Stat. Phys.* **34** (1984), 975.
- [31] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, *Science* **220** (1983), 671–680; see also <http://www.mathworks.com/matlabcentral/fileexchange/10548>.
- [32] G.R. Kneller, V. Keiner, M. Kneller and M. Schiller, *Comp. Phys. Comm.* **91** (1995), 191–214.
- [33] M. Koennecke and P. Arends, *J. Neut. Res.* **4** (1996), 9; see also NeXus, available at: <http://www.nexusformat.org>.
- [34] J. Kohlbacher and I. Bressler, SASfit, PSI, Switzerland, available at: <http://hwr-web.psi.ch/sunsi/SANSSoft/sasfit.html>.
- [35] T.G. Kolda, R.M. Lewis and V. Torczon, *SIAM Review* **45** (2003), 385.
- [36] J.C. Lagarias, J.A. Reeds, M.H. Wright and P.E. Wright, *SIAM Journal of Optimization* **9** (1998), 112.
- [37] K. Lefmann and K. Nielsen, *Neutron News* **10** (3) (1999), 20.
- [38] U. Filges, M. Koennecke, P. Tregenna-Piggott, J. Peters, K. Lieutenants, G. Zsigmond, P. Bentley and E. Farhi, *Journal of Neutron Research* **16** (3) (2008), 97–111.
- [39] J.J. Liang, P.N. Suganthan and K. Deb, Novel composition test functions for numerical global optimization, in: *IEEE Swarm Intelligence Symposium*, 2005, pp. 68–75.
- [40] Matlab Central web site available at: <http://www.mathworks.com/matlabcentral/fileexchange>.
- [41] McStas web site <http://www.mcstas.org>.
- [42] Nelder and Mead, *Computer J.* **7** (1965), 308.
- [43] A. Neumaier, O. Shcherbina, W. Hoyer and T. Vinko, *Math. Prog.* **103** (2005), 335.
- [44] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, 2006.

- [45] T.G. Perring, R.A. Ewings and J. van Duijn, Horace, 2009, available at: <http://horace.isis.rl.ac.uk>.
- [46] D. Richard, M. Ferrand and G.J. Kearley, *J. Neutron Research* **4** (1996), 33–39; see also LAMP, ILL, France, available at: http://www.ill.fr/data_treat/lamp/lamp.html.
- [47] J. Rodríguez-Carvajal, *Physica B* **192** (1993), 55.
- [48] J. Šaroun and J. Kůrda, *Physica B* **234–236** (1997), 1102–1104; see also <http://neutron.ujf.cas.cz/lestrax/>.
- [49] A.R. Secchi and C.A. Perlingeiro, in: *Proc. of XII Congresso Nacional de Matemática Aplicada e Computacional*, Sao Jose do Rio Preto, SP, 1989, pp. 49–52; see also <http://www.mathworks.com/matlabcentral/fileexchange/12776>.
- [50] D.F. Shanno, *Mathematics of Computation* **24** (1970), 647–650.
- [51] B.V. Sheela, *Computer Methods in Applied Mechanics and Engineering* **19** (1979), 99.
- [52] Y. Shi and R.C. Eberhart, A modified particle swarm optimizer, in: *Proc. IEEE Int. Conf. Evol. Comput.*, IEEE Press, Piscataway, NJ, 1998, pp. 69–73.
- [53] N.Z. Shor, *Minimization Methods for Non-Differentiable Functions*, Springer Series in Computational Mathematics, Vol. 3, Springer-Verlag, New York, 1985.
- [54] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.-P. Chen, A. Auger and S. Tiwari, Problem definition and evaluation criteria for the CEC 2005 Special Session on Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, 2005, KanGAL Report #2005005, IIT Kanpur, India.
- [55] D. Wechsler, G. Zsigmond, F. Streffer and F. Mezei, *Neutron News* **11**(4) (2000), 25–28; see also http://www.helmholtz-berlin.de/forschung/grossgeraete/neutronenstreuung/projekte/vites/index_de.html.
- [56] P. Willendrup, E. Farhi and K. Lefmann, *Physica B* **350** (2004), 735.
- [57] P. Willendrup, U. Filges, L. Keller, E. Farhi and K. Lefmann, *Physica B – Cond. Matt.* **385** (2006), 1032.