

Collection SFN 12 (2010)

© Owned by the authors, published by EDP Sciences, 2010

Virtual experiments in a nutshell : simulating neutron scattering from materials within instruments with McStas.

E. Farhi¹, P. Willendrup²

1 Institut Laue Langevin, BP 156, 38042 Grenoble Cedex 9, France

2 Risø National Laboratory, Frederiksborgvej 399 P.O. Box 49 DK-4000 Roskilde

Abstract. We introduce Monte-Carlo methods for neutron scattering with step-by-step examples, using the *McStas* simulation tool. A selection of neutron instrument components are presented, as well as available sample scattering kernels. All these parts are assembled into more advanced instrument models in order to produce so-called virtual experiments, that is simulations which produce results comparable with experiments. Ways to couple such simulations with other simulation software including molecular dynamics are discussed.

1. INTRODUCTION

Building and improving neutron scattering instruments is a long and thorough process. And all existing instruments suffer imperfections, which show-up in sample scattering measurements such as broadening, asymmetry, background and additional contributions. Neutron scattering ray-tracing simulation tools, such as *McStas*, offer the possibility to model many of these effects, which in the end help to understand the instrument pitfalls and improve their usage.

The object of this paper is to present what can be achieved today with Monte-Carlo ray-tracing neutron scattering codes. After introducing the Monte-Carlo method principle (section 2), we recall the definition and applications of virtual experiments (section 3). A short introduction to the *Mcstas* simulation package is presented in section 4. We then demonstrate the use of the various parts that may enter in the description of a complete virtual experiment, including usual neutron optics components and sample models (section 5). Short reminders about the typical use of these components on real instrument are discussed, as well as a schematic description of the underlying physics implemented in sample kernels. These instrument parts are then assembled into simple virtual experiment models (section 6). The most advanced instrument models (section 7) make use of virtual experiments to reveal some of the common artefact inherent to neutron instrumentation, and often enter in the definition of the resolution function. Each concept discussed in this work is demonstrated with a simple *McStas* instrument description example.

2. MONTE-CARLO METHODS FOR NEUTRON SCATTERING

In this section, we present a short introduction to the Monte-Carlo techniques, that is those that use random numbers.

2.1 Definition

The Monte-Carlo technique is simply any process making use of random numbers. In this respect, most processes in nature are random-based: DNA and species evolution within time, traffic jams, weather forecast, flies in the kitchen, stock exchange market, ... Monte-Carlo methods are all around us.

The Monte-Carlo computing method was invented by John von Neumann, Nicholas Metropolis and Stanislaw Ulam [1], and actually named by reference to the well known Casino (*Figure 1*) - from Ulam who was obsessed by randomness in life, and games such as solitaire and poker. This method was first implemented in order to estimate the neutron diffusion in fissionable materials in the frame of the Manhattan Project, and resulted in the first version of Monte-Carlo N-Particle code (MCNP) in 1945.

More specifically for the purpose of this paper, this computing technique may be applied to numerical integration methods which we shall mainly use for neutron scattering simulations, in order *e.g.* to estimate macroscopic measurable values (intensities, spatial distributions, ...) by summing up microscopic quantities.



Figure 1. The Monte-Carlo Casino.
Photograph Source: GALE FORCE Archive

Mathematically, the Monte-Carlo method is an application of the law of large numbers [2,3]. Let $f(u)$ be a finite, continuous, integrable function of parameter u for which an integral estimate is desirable. The discrete statistical mean value of f (computed as a series) in the uniformly sampled interval $a < u < b$ with n values converges to the mathematical f mean value $E(f)$ (also known as the *expectation* of f) over the same interval:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1, a \leq u_i \leq b}^n f(u_i) = \frac{1}{b-a} \int_a^b f(u) du = E(f)$$

In the case where the u_i values are regularly sampled, this corresponds with the well known mid-point integration rule. The Monte-Carlo integration technique considers the case where the u_i values are randomly (but uniformly) sampled, and the random integration step du converges to $(b-a)/n$. As random number generators are not perfect, we rather talk about *pseudo*-Monte-Carlo technique. In principle, Monte-Carlo methods converge to the f function expectation by summing up a large number of infinitesimal random values, in which case we state that the Monte-Carlo estimate is *consistent*.

The Monte-Carlo computer techniques require that the random numbers be unbiased independent variables. The choice of the random number generator is essential to ensure a proper estimate consistency, and the development of the Monte-Carlo techniques have triggered much effort in the computational, pseudo-random number generator algorithms, such as the well known linear congruential generator (using modulus). One way to quantify the efficiency of these random number generators resides in their periodicity, that is the maximum pseudo-random sequence length before it repeats itself, and their maximum dimensionality, that is the maximum sequence length that can be thrown and considered to hold independent equidistributed numbers. Currently, the *Xorshift* [4] and the *Mersenne Twister* [5] algorithms are among the most efficient pseudo-random number generators.

2.2 Error estimate and convergence

Even though the convergence of the Monte-Carlo estimate for the finite continuous integrable function f is guaranteed for infinite number of random evaluations n , it may be appropriate to estimate the associated error for finite values of n . The variance of the estimator is defined as the averaged square deviation of the $f(u_i)$ values from its expectation $E(f)$

$$V(f) = E[f - E(f)]^2$$

The law of large numbers and the central limit theorem state that the Monte-Carlo estimate follows a normal distribution when n is large. In this case, the standard deviation of the estimate is

$$\sigma = \frac{\sqrt{V(f)}}{\sqrt{n}}$$

which indicates that the error on the estimate follows a $n^{-1/2}$ law, independent of the dimensionality d of the integration (that is the number of independent variables to integrate in order to obtain the integral estimate). This result is essential,

compared with other integration techniques, such as the trapezoidal rule, which behaves as $n^{-2/d}$ and the Simpson's rule ($n^{-4/d}$). In a few words, the Monte-Carlo technique is guaranteed to converge faster than the trapezoidal and the Simpson's rule for dimensionality d larger than 4 and 8 respectively, and more generally any usual quadrature integration rule will be slower than the Monte-Carlo estimate at some point.

In the case of the *McStas* Monte-Carlo neutron scattering ray-tracing software [6,7], we use a $d=10$ phase space (position, velocity, spin, and time) from which we integrate measurable quantities. In this space, we have estimated the number of random numbers to cast in order to obtain a given accuracy in the integral $E(f)$. Basically, throwing 10^6 neutron events ensures an error bar within a percent for most simulations. Such simulations usually run within a few seconds. The most advanced examples in this work may require up to 10^8 neutron events which then produce accurate results even for low probability processes (e.g. scattering in materials) within minutes.

Records in $E(f)$	10^3	10^4	10^5	10^6	10^7
Accuracy $\sigma/E(f)$ [%]	10	2,5	1	0,25	0,05

Table 1. Uncertainty estimate, in percents, as a function of the number of statistical events used to compute an integral with *McStas*

2.3 Variance reduction techniques: improving the computation efficiency

Even though the Monte-Carlo techniques all use random numbers, there are many ways to implement the method.

Suppose a given physical process has a probability p to occur, and thus be counted in an integral. One easy solution is to cast n equally distributed random numbers $0 < \pi < 1$, and only count events that satisfy $\pi < p$. After n initial random numbers, the $E(f)$ integral only counts np events and the relative standard deviation is $\sigma \propto 1/\sqrt{np}$. Clearly this is not efficient, especially when p is small. We may then label this method as a 'hit or miss' algorithm.

An other way to obtain exactly the same estimate is to assign a statistical weight p_i to each event (and in the previous case $p_i = p/n$), and count them all in a weighted expression of $E(f)$

$$E(f) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1, a \leq u_i \leq b}^n p_i f(u_i) \quad \text{with} \quad \sum_{i=1}^n p_i = p$$

which in this case uses the n events, and the standard deviation is $\sigma \propto 1/\sqrt{n}$. This is an extremely common trick in Monte-Carlo integration, referred to as *importance sampling*, and it actually lowers random number casting. For instance, most of the absorption estimate in *McStas* is taken into account that way.

Among the other variance reduction techniques, one is to identify u_i space regions that have larger weight in the final result. Suppose we aim at estimating the scattered intensity from a sample, hitting a detector. The integral $E(f)$ may then be written as

$$E(f) = E(f)_{\text{events that will probably hit the detector}} + E(f)_{\text{events that may not hit the detector}}$$

It is reasonable to assume that the second term is substantially smaller than the first one in the total result. If we now choose to cast more random numbers in the first term than in the second, we shall probably improve the statistical accuracy on the result. This is known as *stratified sampling* which favours regions of the phase space *w.r.t.* other parts which are assumed to contribute less to the result. This trick requires that we are able to, in a way, define the high importance regions, which implies to define a choice criteria (hit or miss sample and detector). In the sample scattering example cited above, these events could simply be defined as neutron rays that actually hit the sample and thus have a chance to reach the detector, whereas the second part could be those that miss the sample, and mostly contribute to the background.

Finally, up to now, we essentially use equi-distributed random numbers (that is regularly sampled within a range). In some cases, we may directly estimate the probability distribution assigned to a given process which will count in the result. Then, there are ways to modify the initial, flat, equi-distributed, random number generator into a more efficient focused distribution. Coming back to the 'hit or miss' example above with probability p , this boils down to choosing more random numbers π that satisfy $\pi < p$ than $\pi > p$. This is known as *adaptive variance reduction*.

We encourage the reader to refer to James [2] for a detailed review on the Monte-Carlo method.

3. VIRTUAL EXPERIMENTS

3.1 Definition of virtual experiments

Following [8], 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 [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.

3.2 Applications of virtual experiments

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 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 will effectively be measurable within an experiment. 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 contributions from 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.

4. INTRODUCTION TO MCSTAS

The *McStas* software package [6,7,9] is a European effort to bring an open-source common frame-work for modelling of all kinds of neutron scattering instruments (diffractometers, spectrometers, reflectometers, small-angle, back-scattering,...) for both continuous and pulsed sources. It is based on a meta-language specially designed for neutron scattering simulation. Instrument descriptions are written in this language by users and automatically translated into efficient simulation codes in ANSI-C. The present version *McStas 1.12b* (July 15th, 2010) supports both continuous and pulsed source instruments, and includes a library which contains around 110 components and 70 instrument examples.



Figure 2. The McStas main interface 'McGUI'.

4.1 Installing McStas

The *McStas* software runs on all common operating systems, including Windows XP, Vista and 7, Linux flavours, and Mac OSX. Installer packages are available [9]. Currently, *McStas* only requires a C compiler, but higher level user interfaces make use of Perl, Perl-Tk, PGPLOT, and Perl-DL. We optionally recommend to install Scilab, a VRML viewer, MPI libraries for multi-processor and cluster support, and Matlab. Support for the common X-ray and neutron data format *Nexus* [10] is also available.

4.2 Starting a simulation with McStas

Starting the *McStas* software will typically result in showing up the main '*McGUT*' interface (`mcgui` command), as shown in *Figure 2*. We then recommend to start by selecting an instrument example in the *Neutron Site* menu, such as the *Templates/templateDIFF* one, which is a standard diffractometer model. Pressing the **Run** button will assemble the instrument model and request instrument parameters, which are for this example the incident neutron wavelength λ (defaulting to 1 Å), the monochromator lattice spacing *DM* (defaulting to 3.355 Å) and curvature *RV*, the sample powder structure definition file *Powder* (defaulting to Sapphire), and instrument distances *L1*, *L2* and *L3*. For a start, just press the *Start* button and wait for the simulation completion, which should be shorter than a minute. From the main interface, press the **Plot** button in order to display the simulation results, which are in this case some histograms built from the neutron beam distribution at various places in the instrument. In our case we shall focus on the two last monitors, which present the angle dispersed diffractogram, as well as a 2D detector showing portions of the sample scattering Debye-Scherrer cones [14], as shown in *Figure 3*. You may of course change the instrument simulation parameters from the *Run Dialogue* without actually touching the model itself, for instance the incoming neutron wavelength or sample structure file.

4.3 Viewing the instrument model

Pressing again the *Run* button from the main interface, but changing the *Simulate* mode into the *Trace 3D* mode with a *HTML/VRML* plotter format (see *Figure 4*), the *Start* button generates an interactive 3D view (see *Figure 5*). The corresponding *McStas* instrument description will be discussed in section 5.6. It can be displayed by pressing the *Edit/New* button.

4.4 Getting help and going further

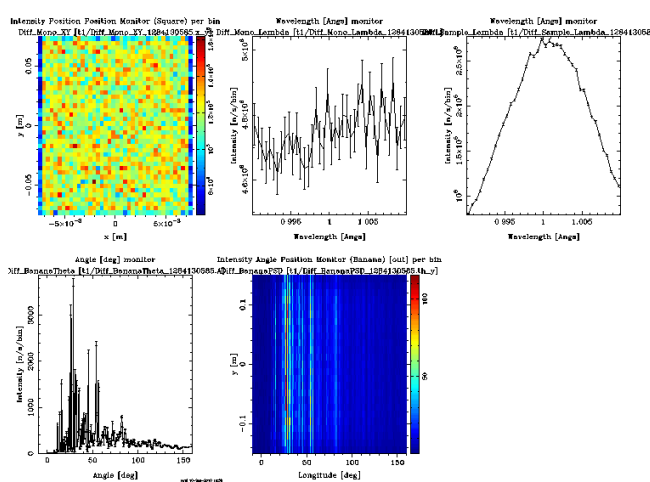


Figure 3. Results from a typical diffractometer model simulation using *McStas*. The two lower plots show the angular diffractogram, and the 2D detector.

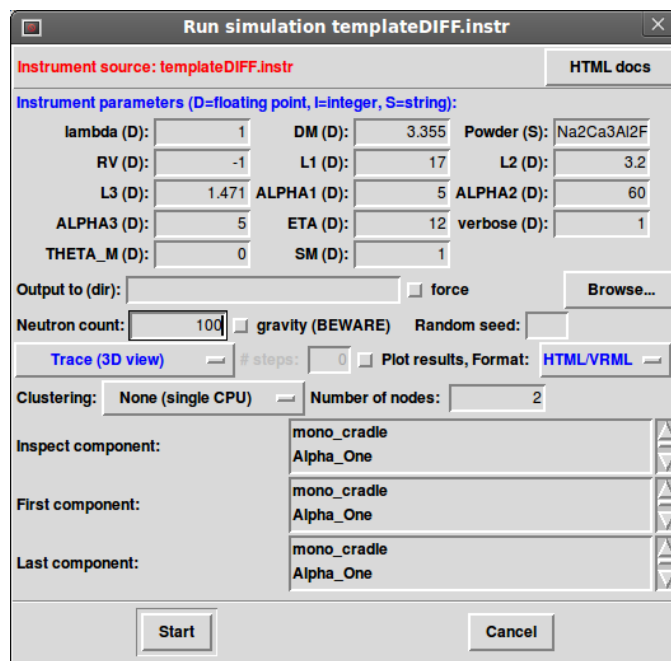


Figure 4. The '*Run simulation*' dialog set to the '*Trace 3D*' mode in order to visualize the model geometry with VRML.

The *Help* menu on the right side of the main *McStas* interface links to the *User Manual* (which indicates how to use *McStas*, write instruments and new components), the *Component Manual* (which lists most of the available components), and the *Component Library Index* which shows a compact list of all available data files, components and example instruments. We also encourage users to register on the mcstas-users@mcstas.org mailing list [9], to which they can post questions and follow discussions with many other *McStas* users.

When you feel confident with *McStas*, using existing example instruments and components, you can start to build your own instrument descriptions (as shown below in sections 5.5, 5.6, and 6) and eventually modify existing or implement new components. We shall not discuss this latter matter in the frame of this paper, but rather point you to the *McStas User Manual*. Feel free to send your new data files, instruments and components to the mailing list and the *McStas* core team so that they can be included in further releases and benefit the whole *McStas* and neutron community.

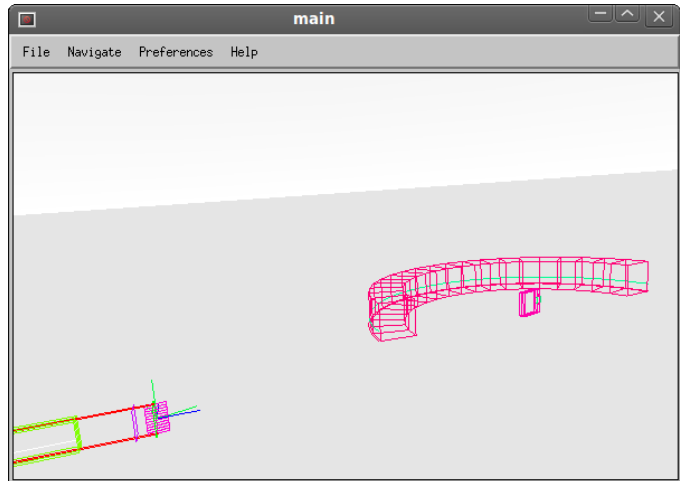


Figure 5. Instrument model geometry as shown with *McStas/McDisplay*. The neutron beam tube is shown in green, followed by the multi-blade monochromator which deflects the neutron to a collimator just before the sample. A radial collimator precedes a banana shaped detector (in magenta).

5. VIRTUAL EXPERIMENT COMPONENTS

In the following sections, we introduce a number of common instrument simulation components, to be used with *McStas 1.12b*. In each case, usage examples are listed, as well as the corresponding model geometry, when appropriate. We present some simple neutron source models, most common neutron optics, samples and ideal detectors. Each example is fully described as a plain text file, in which we have highlighted the *McStas* grammar keywords in blue, the *McStas* components in green, the instance names defined by the user in red, the *McStas* pre-defined symbols in italic, and the pure C keywords in bold. To use these examples, you may type them into instrument description files (plain text format with *.instr* extension), open them with *McStas/McGUI* and press the *Run* button. The usual simulation time is within seconds (with 10^6 neutron events), and possibly reaches a few minutes when higher statistics are required.

In the following sections, a number of standard notations are used: λ for wavelength [\AA], E and ω for the energy [meV], k for the momentum [\AA^{-1}] and v for the velocity [m/s]. These quantities are related by the simple relations

$$E_i = 81.804/\lambda^2 = 2.072 k_i^2, k = 2\pi/\lambda, v = 3956/\lambda \text{ for neutrons.}$$

The coordinate system used in *McStas* defines the Y axis as vertical, the X axis left bound when looking forward, and the Z axis in the forward direction of propagation. This frame is used for positioning and rotating instrument components in space. In the latter case, rotations apply around each axis, in the order X, Y and Z.

This work considers neutrons in the cold-thermal energy range, neglecting higher energy nuclear physics processes (fission, resonances, Compton scattering) [19]. No polarised neutron component is considered, even though *McStas* provides a set of optic and sample components.

5.1 Sources

Source components generate neutron particles, that is a random distribution in space (source volume/area), velocity (neutron energy), time (especially for pulsed sources) and possibly spin. Additionally, the particle Monte-Carlo weight integrates as the total source flux. We shall not go into the details of the neutron generation in sources, but only consider on the simulation side that neutron events are evenly generated on a surface, with given energy, divergence and time distributions.

Component name	Description
<i>Source_simple</i>	A simple flat spectrum source, steady state.
<i>Source_Maxwell_3</i> and <i>Source_gen</i>	A black body steady state source. Parameters for the ILL, and PSI are available.
<i>Moderator</i>	Simple pulsed source
<i>ESS_moderator_long</i>	An ESS parametrized long-pulsed source
<i>SNS_source</i>	An MCNP/X based SNS (USA) short-pulsed source
<i>ISIS_moderator</i>	An MCNP/X based ISIS (England) pulsed source

Table 2. A short list of some common neutron source components available in McStas.

McStas provides a set of steady state sources *e.g.* for reactor based instruments, as well as some other components more suited for pulsed sources (refer to **Table 2**). As a start, we recommend new *McStas* users to use the ***Source simple*** component. The more advanced *Source_gen* component can also read tabulated spectrum flux. The next sections demonstrate a number of usage examples of source components. Detailed source models may be imported from neutron transport codes such as MCNP and FLUKA, as explained in section **8.1**.

5.2 Neutron optics

Once neutron particles have been emitted by a neutron source model, they propagate to other model components which we label as neutron optics, except for samples and detectors (see dedicated sections **5.3** and **5.4**). We only present the most common optics components, but others are available from the *McStas* library (*e.g.* refractive lenses and prisms, statistical choppers, filters, polarized components).

5.2.1 Guides

Apart from beam tubes, which have non reflective surfaces and thus effectively act as simple slits, guides enable to transport neutron beams far away from the reactor. State of the art multi-layer coated guides have a very good transmission and also drastically lower the neutron source background noise level on the distant instruments by transporting neutrons far away from the neutron source.

There are many ways to model neutron guides, which essentially depend on the complexity of the guide geometry. Most of the available guide components model straight guide sections, including the simplest and recommended ***Guide*** component. A curved guide may be built by appending a series of *Guide* components along a curved line, such as in the *Example 1*. A small gap *d* in between straight elements has been added to permit a small rotation without actually overlapping adjacent components. Running this instrument description will provide a single neutron beam image at the end of the guide. It is intended to be used as an initial part of an instrument. The *Figure 6a* shows the guide geometry (see section **4.3**), as produced by running the model in the *Trace* mode (see section **4.3**) or issuing the `mcdisplay curved_guide.instr R=2700` command. The *Figure 6b* shows the simulation results recorded by the monitor component. In practice, a guide with a constant width *w* [m], a Ni-reference coating *m* value, and a curvature radius *R* [m] should have a minimum length $L = \sqrt{8wR}$ [m] to bring the neutron beam out of direct sight of the source. The neutrons are then transmitted when their wavelength is greater than $\lambda_c = 180 / (0.1 * m \pi) * \sqrt{2w/R}$ [Å].

```

/*****
* Instrument: curved_guide.instr
* %Description
* A curved guide example made of straight elements
*
* %Parameters
* lambda: [Angs] neutron wavelength
* dlambda: [Angs] neutron wavelength spread, half width
* L:      [m] guide total length
* R:      [m] curvature radius
*****/
DEFINE INSTRUMENT curved_guide (lambda=4, dlambda=3.5, R=2700, L=14)
DECLARE %{
    double n=10; /* [1] number of straight guide elements */
    double d=1e-3; /* [m] spacer in between adjacent straight elements */
%}
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.03, xw=0.03, yh=0.2,
                                Lambda0=lambda, dLambda=dlambda)

    AT (0,0,0) ABSOLUTE
COMPONENT CG_1 = Guide(l=L/n, w1=0.03, h1=0.2, w2=0.03, h2=0.2)
    AT (0,0,1) RELATIVE PREVIOUS ROTATED (0, (L/n+d)/R*180/PI, 0) RELATIVE PREVIOUS
COMPONENT CG_2 = COPY(CG_1)
    AT (0,0,L/n+d) RELATIVE PREVIOUS ROTATED (0, (L/n+d)/R*180/PI, 0) RELATIVE PREVIOUS
/* ... make it so with a total of n=10 elements */
COMPONENT CG_10 = COPY(CG_1)
    AT (0,0,L/n+d) RELATIVE PREVIOUS ROTATED (0, (L/n+d)/R*180/PI, 0) RELATIVE PREVIOUS
COMPONENT CG_Mon= Monitor_nD(xwidth=0.03, yheight=0.2, options="x y")
    AT (0,0,L/n+d) RELATIVE PREVIOUS
END

```

Example 1. A curved guide example made of straight elements. Simple bits of C language are used in the DECLARE block to define internal constant values, n and dt . The default guide coating is Nickel ($m=1$). To assemble this Example, type the description in the McGUI editor, save it into the `curved_guide.instr` file, and press the *Run* button.

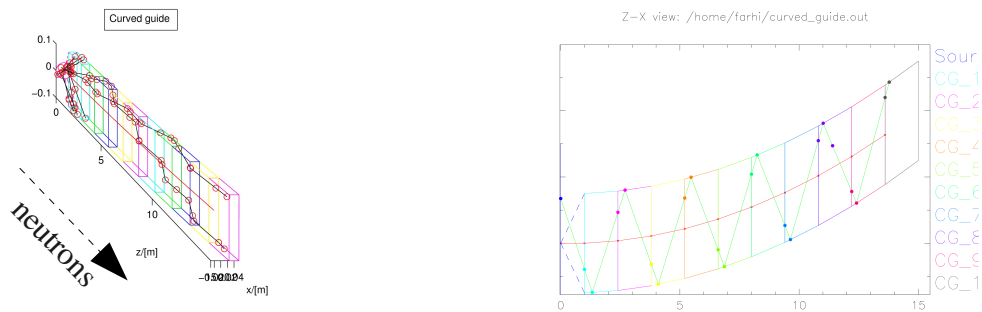


Figure 6a. The curved guide *Example 1* geometry shown with Matlab[®] (left) and PGPLOT (right) plotters. Some neutron trajectories are displayed as a series of straight lines. These views are obtained by starting the simulation in *Trace* mode.

In principle, this type of polygonal guide may model most geometries, including ballistic, elliptic and parabolic guides made of flat surfaces. More advanced geometries may be described using for instance the *Guide_gravity*, the *Guide_channeled*, the *Guide_curved*, and the *Guide_tapering* components.

5.2.2 Collimators

The primary role of a collimator is to restrict the neutron beam divergence, that is absorb all neutrons with trajectories inclined by more than a given angle from the main axis. Only neutrons which do not hit the collimator channel sides will remain, with a divergence lower than w/L , with w being the channels width and L their length. Most neutron instruments tend to restrict the incoming neutron divergence in the horizontal plane with such components, but usually leave the vertical divergence unaffected in order not to reduce too much the neutron intensity beyond.

We present in *Example 2* a typical use of the *Collimator_linear* component, to be part of a *McStas* instrument simulation description. The *Guide_channeled* and the *Guide_gravity* components may also be used to model collimators. Radial collimator models, such as the *Collimator_radial* component, are to be used *e.g.* after the sample position. Such collimators are usually made on real instruments from a set of linear component modules stacked side by side. Both geometries are shown in *Figure 7*.

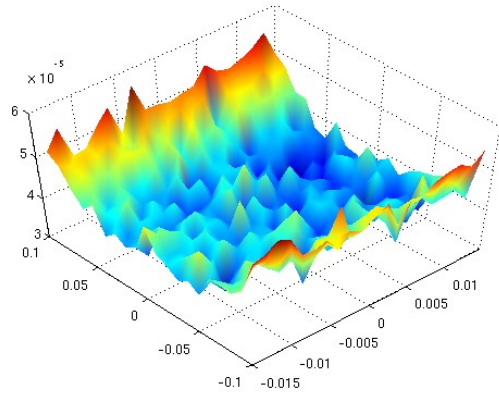


Figure 6b. Results from the *Example 1*, obtained from the neutron beam cross section at the end of the curved guide. Obtained by running the simulation and pressing the *Plot* button. Intensity is shown as color scale from blue (low) to red (high).

```

/* ALPHA: [arc min] collimator divergence. One arc minute is 1/60 degree */
DEFINE INSTRUMENT ... (... , ALPHA=60)
...
TRACE
...
COMPONENT linear = Collimator_linear(
  xwidth = 0.07, yheight = 0.15, length = 0.70, divergence = ALPHA)
AT (0,0,0.05) RELATIVE PREVIOUS
...

...
COMPONENT radial = Collimator_radial(
  radius=0.3, yheight = 0.15, length = 0.40, divergence = ALPHA,
  theta_min=10, theta_max=140)
AT (0,0,0) RELATIVE PREVIOUS
...

```

Example 2. Linear (top) and radial (bottom) collimator instances controlled with an ALPHA instrument parameter.

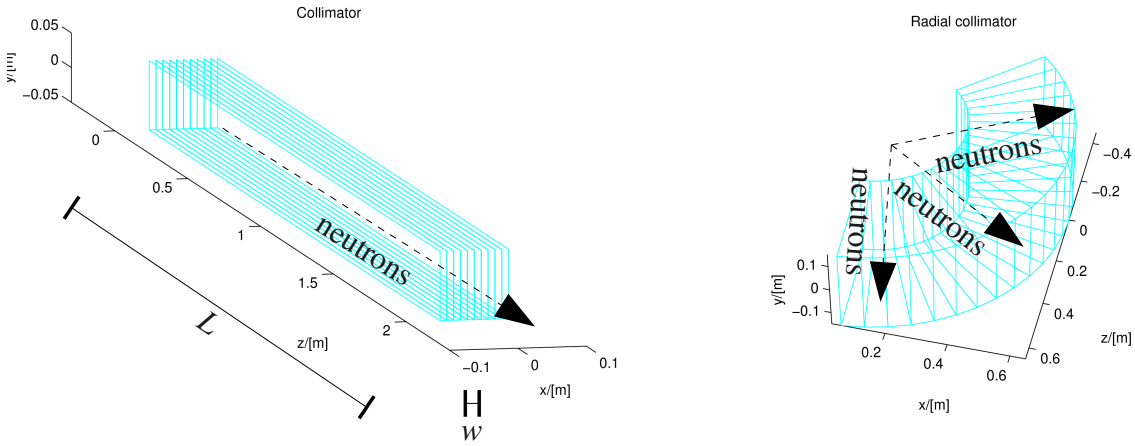


Figure 7. Linear (left) and radial (right) collimator components geometry from the *Example 2*. The collimator slits have a width w , and a length L .

5.2.3 Choppers

One common way to select a neutron energy is to select its speed when passing in between two moving apertures, separated by a distance d . Let us consider a neutron that passes the initial slit ; it will only pass the second slit if this latter is opened after a time delay Δt which is related with the neutron speed $v=d/\Delta t$. Such a condition may be obtained with two disks rotating with a frequency f (in turns per second [Hz]), with the second disk shifted by a phase angle $\phi=2\pi \Delta t f$ with respect to the first. We then determine the neutron wavelength λ^{DC} selected by the two disk choppers as :

$$\lambda_{[Angs]}^{DC} = \frac{3956}{v_{[m/s]}} = \frac{3956 \phi_{[rad]}}{2\pi f_{[Hz]} d_{[m]}} .$$

The selected wavelength band of the pulse, which is ideally triangular, will get narrower as the rotation frequency increases, and the transmitted intensity decreases. The time window selected by the choppers is $\Delta t=\phi/2/\pi f$. In order to avoid multiple order transmissions (that is phase angles α that are shifted by $2\pi.n$, with $n>1$) additional disks are usually required, resulting in systems with often more than 4 rotating disks.

Simulating a disk chopper system consists in positioning at least two **DiskChopper** components. The steady state components listed in **Table 2** do not define the initial neutron event time t the same way for continuous (time is set to $t=0$) and pulsed sources (time t spans over the pulses). In order to adapt the **DiskChopper** instance to these sources components, the instrument description should specify the first **DiskChopper** parameter $IsFirst=1$ flag for steady-state sources, and leave it to $IsFirst=0$ on pulsed sources, as shown in *Example 3* and *Figures 8a* and *8b*. The last component is a time-of-flight monitor which determines itself the better suited time window.

```

/*****
* Instrument: disk_choppers.instr
* %Description
* A disk chopper example made of two instances. The phase is automatically set.
*
* %Parameters
* lambda: [Angs] neutron wavelength selected by the chopper system
* frequency: [Hz] chopper rotation frequency, omega=2*PI*frequency, rpm=60*frequency
* d_cc: [m] distance between the two choppers. Chopper opening slits is 10 deg here.
*****/
DEFINE INSTRUMENT disk_choppers (lambda=4, frequency=200, d_cc=5)

TRACE
COMPONENT Source = Source_simple(dist=1+d_cc, radius=0.03, xw=0.03, yh=0.2,
                                Lambda0=lambda, dLambda=lambda*0.01)

    AT (0,0,0) ABSOLUTE
COMPONENT Chop1 = DiskChopper(omega=2*PI*frequency, R=0.3, h=0.2, theta_0=10, n=1, IsFirst=1)
    AT (0, 0, 1) RELATIVE PREVIOUS
COMPONENT Chop2 = COPY(Chop1)(IsFirst=0, phi_0=2*PI*frequency*d_cc*lambda/3956*RAD2DEG)
    AT (0, 0, d_cc) RELATIVE Chop1
COMPONENT CG_Mon= Monitor_nD(xwidth=0.03, yheight=0.2, options="auto time")
    AT (0,0,0.1) RELATIVE PREVIOUS
END
    
```

Example 3. A simple disk chopper system after a continuous source. A similar instrument using a pulsed source such as the *ESS_moderator_long*, would require *IsFirst=0*. This instrument description makes use of the *Monitor_nD* with its automatic histogram limits feature for the neutron time parameter.

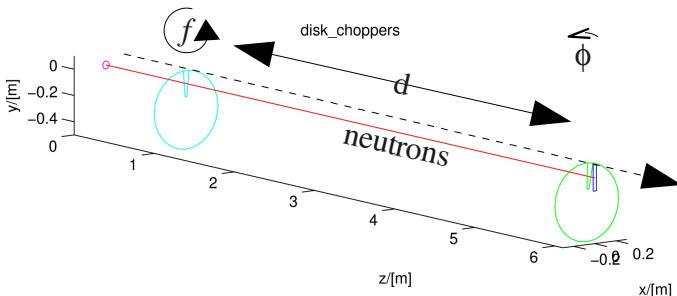


Figure 8a. A double chopper spectrometer from the *Example 3*. Disks are de-phased by angle ϕ , rotating with frequency f , separated by distance d .

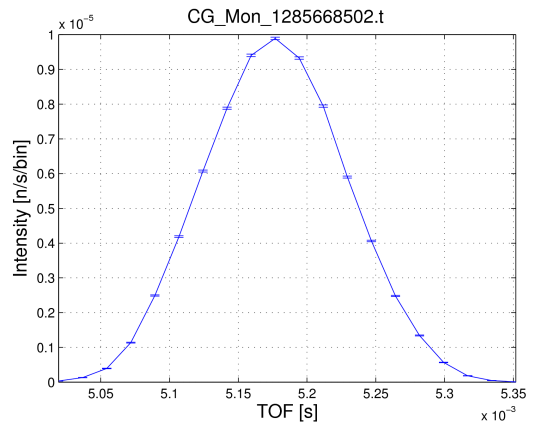


Figure 8b. Time distribution from the double chopper spectrometer operating at $\lambda=4 \text{ \AA}$.

The Fermi choppers are a more compact alternative to disk chopper arrangements. They consist in a rotating, short collimator. In order for neutrons to be transmitted, they have to pass the entry window of the collimator, and propagate directly to the exit window without touching the collimator side walls. All of this takes place in a rotating frame, which makes the model slightly more complex [11,12].

The neutron wavelength λ^{FC} selected by the Fermi chopper can be found to be close to :

$$\lambda_{[Angs]}^{FC} = \frac{3956 w_{[m]}}{2 \pi f_{[Hz]} L_{[m]}^2}$$

with w being the slit width, and L its length. The opening time of the Fermi chopper is about $\Delta t = w/L/2/\pi/f$ and the beam divergence is restricted to w/L . The Fermi chopper wavelength resolution will get narrower as the frequency increases and the slit width decreases, which also leads to a decrease in intensity.

```

/*****
* Instrument: fermi_chopper.instr
* %Description
* A Fermi chopper example. The rotation frequency is computed automatically, and the initial
* time distribution is set by an EXTEND after the source. Parameters from IN6@ILL Fermi Chopper.
*
* %Parameters
* lambda: [Angs] neutron wavelength selected by the chopper system
*****/
DEFINE INSTRUMENT fermi_chopper (lambda=4)
DECLARE %{
  double w = 0.064/200; /* width of a single slit in the Fermi chopper package */
  double L = 0.012; /* length of the Fermi chopper package */
  double frequency, dt; /* declare internal variables to be used afterwards */
%}
INITIALIZE %{
  frequency = 3956*w/(2*PI*lambda*L*L);
  dt = w/L/2/PI/frequency; /* Fermi chopper time window */
  printf("%s: frequency=%g [Hz], time window dt=%g [s].\n", NAME_CURRENT_COMP, frequency, dt);
%}

TRACE
COMPONENT Source = Source_simple(dist=1, radius=0.03, xw=0.064, yh=0.05,
                                Lambda0=lambda, dLambda=lambda*0.01)

  AT (0,0,0) ABSOLUTE
  EXTEND %{
    double v = sqrt(vx*vx+vy*vy+vz*vz); /* neutron velocity */
    t = randpml()*dt-1/v; /* set neutron event time in Fermi chopper
                          time window and shift to Fermi chopper at 1 [m] */
    p = p * dt; /* correct intensity which is set for 1 [s] */
  %}
COMPONENT FC = FermiChopper(nu=frequency, w=w, height=0.05, Nslit=200, length=L)
  AT (0, 0, 1) RELATIVE PREVIOUS
COMPONENT FC_Mon= Monitor_nD(xwidth=0.064, yheight=0.05, options="auto time")
  AT (0,0,0.1) RELATIVE PREVIOUS
END

```

Example 4. A simple Fermi chopper setup, which computes automatically the rotation frequency, and sets time for neutrons to reach the chopper time window. This example makes use of the DECLARE and INITIALIZE blocks to handle internal variables, and of the EXTEND block to act on the neutron event time t and weight p .

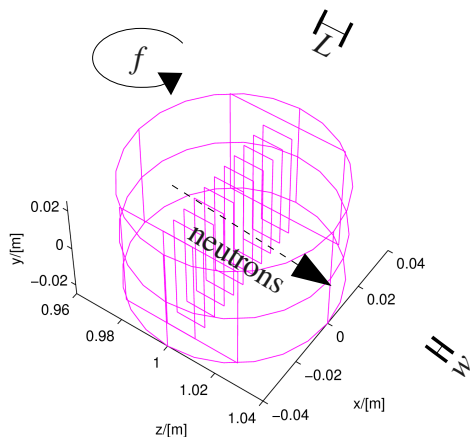


Figure 9a. A Fermi chopper geometry from the *Example 4*. Slit width is w , length L , rotation frequency f .

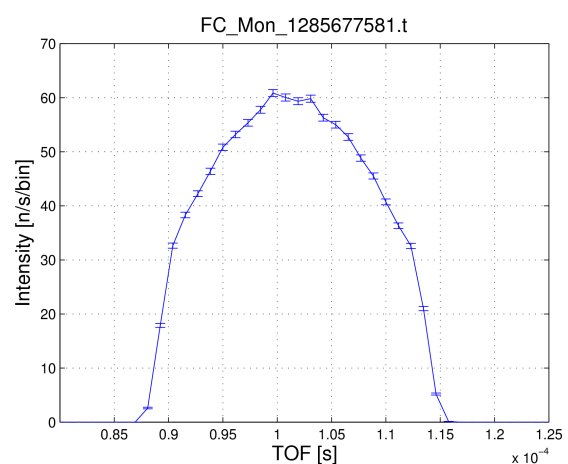


Figure 9b. Time distribution from the Fermi chopper spectrometer, at $\lambda=4$ Å.

The *Example 4* shows a simple use of the *FermiChopper* component, with some additional *McStas* grammar features, such as the component instance extension with the EXTEND keyword. The geometry, as well as the obtained result are shown in *Figures 9a* and *9b*.

It appears that a disk chopper system may be replaced with an equivalent Fermi chopper when the condition $L^2/w = d/\phi$

is satisfied. Both devices are mainly used on time-of-flight spectrometers in order to obtain a highly monochromatic neutron beam, with typically a wavelength range of a few percent around the central transmitted wavelength.

5.2.4 Velocity selectors

A chopper-like mechanical device is obtained with velocity selectors, which are used to select neutrons by transmitting only a given energy. They consist in a rotating drum, main axis usually parallel to the neutron beam axis, with its circumference covered with radial twisted blades, as shown in *Figure 10a*. For neutrons not to be absorbed, they have to pass any of the entry windows, remain in the selected channel in between the absorbing twisted blades (angle ϕ) while propagating (length d), and exit the rotating drum (frequency f).

```

/*****
* Instrument: velocity_selector.instr
* %Description
* A velocity selector example. From D11@ILL selector (EADS Astrium).
*
* %Parameters
* lambda:    [Angs] neutron wavelength selected by the velocity selector
* phi:      [deg] velocity selector twist angle
* d_vs:     [m] velocity selector rotating drum length
*****/
DEFINE INSTRUMENT velocity_selector (lambda=4, phi=48.3, d_vs=0.25)

TRACE
COMPONENT Source = Source_simple(dist=1, radius=0.03, xw=0.03, yh=0.05,
                                Lambda0=lambda, dLambda=lambda*0.15)
  AT (0,0,0) ABSOLUTE
COMPONENT VS = V_selector(width=0.03, height=0.05, l0=0.30, r0=0.12, phi=phi,
                          l1=d_vs, tb=0.0004, rot=3956*phi*DEG2RAD/(2*PI*lambda*d_vs)*60, nb=72)
  AT (0, 0, 1) RELATIVE PREVIOUS
COMPONENT VS_Mon= Monitor_nd(xwidth=0.064, yheight=0.05, options="auto wavelength")
  AT (0,0,d_vs+0.1) RELATIVE PREVIOUS
END

```

Example 5. A simple velocity selector setup, which computes automatically the rotation frequency.

It can be seen that a velocity selector follows the same behavior as a double chopper system, but with many input channels. It does not select monochromatic pulses, but ensures a continuous monochromated beam, better suited for steady-state neutron sources. The transmitted neutron wavelength λ^{VS} follows a similar rule as the double chopper [13]

$$\lambda_{[Angs]}^{VS} = \frac{3956}{v_{[m/s]}} = \frac{3956 \phi_{[rad]}}{2\pi f_{[Hz]} d_{[m]}}$$

where f is the rotation frequency, d is the length of the drum and ϕ is the twist angle. The *Example 5* shows a simple implementation of the *V_selector* component, and the resulting wavelength range, typically 10% around λ^{VS} . These neutron optics devices are mainly used on small angle neutron scattering instruments which do not require a highly monochromatic neutron beam, and as pre-monochromators in front of *e.g.* monochromators (see below).

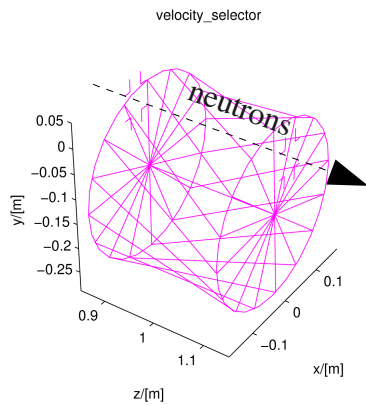


Figure 10a. A velocity selector geometry from *Example 5*, showing drum length d , rotation frequency f and twist angle ϕ .

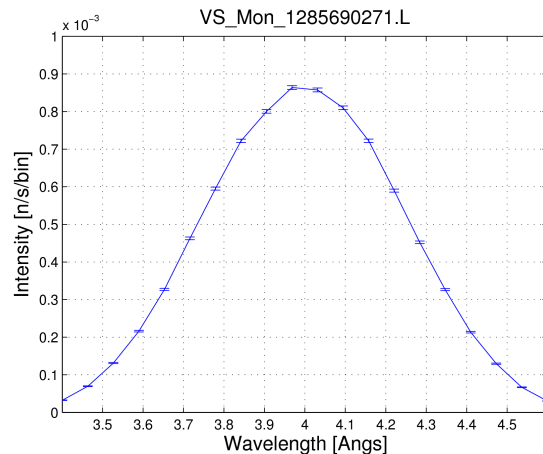


Figure 10b. The wavelength distribution around $\lambda=4 \text{ \AA}$ obtained after a velocity selector, on a continuous neutron source.

5.2.5 Monochromators

The previous monochromating neutron optics all use movable parts, filtering the neutron velocity with a propagation time in between slits. The crystal monochromators are static devices, which select the neutron energy from the reflection (constructive interferences) on atomic planes following the Bragg's law:

$$n \lambda = 2 d \sin \theta$$

where λ is the incident neutron wavelength, d is the distance which separates the diffractive atomic planes, θ is the incidence angle w.r.t. the plane surface, and n is the order of the reflection. In practice, as a crystalline material contains many atomic planes, a polychromatic neutron beam will be reflected in many directions, providing narrow scattered beams each of which will correspond with a single diffracted wavelength.

```

/*****
* Instrument: monochromator_simple.instr
* %Description
* A monochromator example.
*
* %Parameters
* lambda:    [Angs] neutron wavelength selected by the monochromator
* DM:       [Angs] Monochromator d-spacing, default is 002 Graphite
*****/
DEFINE INSTRUMENT monochromator_simple (lambda=2.36, DM=3.355)
DECLARE %{
  double theta; /* half take-off, used further in the instrument TRACE */
%}
INITIALIZE %{
  theta = asin(lambda/DM/2)*RAD2DEG; /* also called A1 on triple-axis machines */
  printf("%s: 2*theta=%g [deg] take-off angle\n", NAME_CURRENT_COMP, 2*theta);
%}

TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.01, xw=0.01, yh=0.01,
                                Lambda0=lambda, dLambda=lambda*0.04)

  AT (0,0,0) ABSOLUTE
COMPONENT MC = Monochromator_curved(width = 0.10, height = 0.10, DM=DM)
  AT (0,0,0.5) RELATIVE PREVIOUS ROTATED (0,theta,0) RELATIVE PREVIOUS
COMPONENT MC_out= Arm() AT (0,0,0) RELATIVE PREVIOUS ROTATED (0,theta,0) RELATIVE PREVIOUS
COMPONENT MC_Mon= Monitor_nD(xwidth=0.01, yheight=0.01, options="auto wavelength ; x")
  AT (0,0,0.5) RELATIVE PREVIOUS
END

```

Example 6. A simple monochromator setup, which computes automatically the take-off angle.

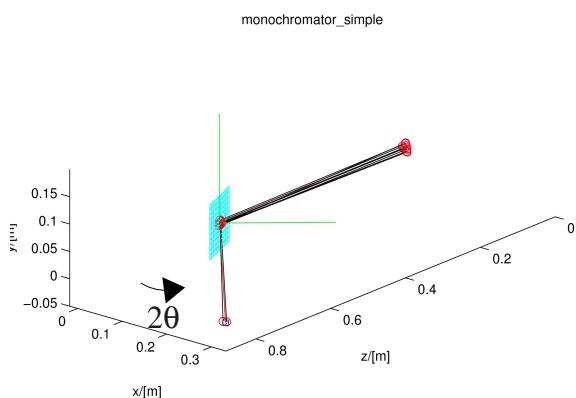


Figure 11a. Geometry of the *Example 6* simple monochromator setup, showing a few diffracted neutron trajectories, with take-off angle 2θ . The *Example 7* has the same geometry.

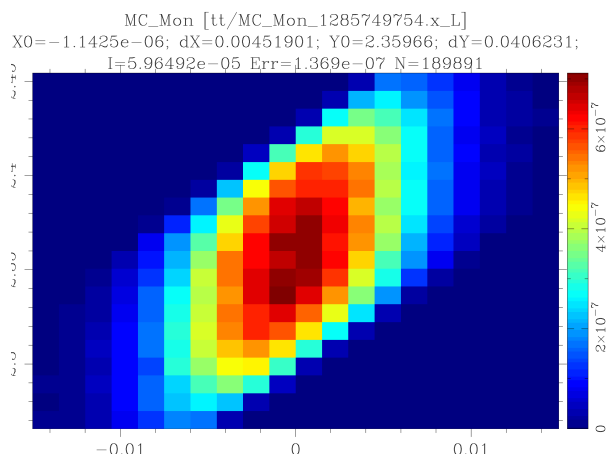


Figure 11b. Wavelength distribution obtained by diffraction on a monochromator at $\lambda=2.36$ Å, as a function of the transverse position in the diffracted spot. Intensity is shown as color scale from blue (low) to red (high).

In the *Example 6*, we use a flat instance of the *Monochromator_curved* component for which the rotation angle θ is computed automatically from the desired wavelength. The model geometry is shown in *Figure 11a*, and the position-wavelength distribution in *Figure 11b* for $\lambda=2.36$ Å neutrons and a (002) graphite reflection. This shows that the monochromating effect depends on the position in the diffracted beam, where one side is more energetic than the other. A better neutron beam focusing could be achieved, *e.g.* to increase the intensity on the sample position, by specifying the curvature radius RH and RV parameters in the component instance (see section 6.2 below).

```
TRACE
...
COMPONENT SX = Single_crystal(
  xwidth = 0.002, yheight = 0.1, zthick = 0.1,
  mosaic = 30, reflections = "C_graphite.lau", barns=0,
  ax = 0,    ay = 2.14, az = -1.24,
  bx = 0,    by = 0,    bz = 2.47,
  cx = 6.71, cy = 0,    cz = 0,
  absorption = 0.014, incoherent = 0.004)
AT (0,0,0.5) RELATIVE PREVIOUS ROTATED (0,theta,0) RELATIVE PREVIOUS
...
```

Example 7. A more advanced monochromator setup, which also takes into account multiple scattering and blade thickness. This example is derived from *Example 6*.

The *Monochromator_curved* component only considers a single d -spacing in the diffracting crystal, infinitely thin blades. However it does take into account multiple diffraction order n , which corresponds to d/n atomic plane spacing in the Bragg law. In order to consider all possible reflections in the crystal, we may use a more advanced single crystal model, such as in *Example 7*, derived from *Example 6* by replacing the *Monochromator_curved* component instance by a *Single_crystal* one. A data file containing the structure factors of graphite is used instead of a single d -spacing value. For more details about this component, refer to the section 5.3.3.

Monochromators are essentially used on diffractometers, back-scattering and crystal time-of-flight spectrometers, and triple-axis inelastic spectrometers. The selected wavelength range is typically about 1 % around the diffracted wavelength.

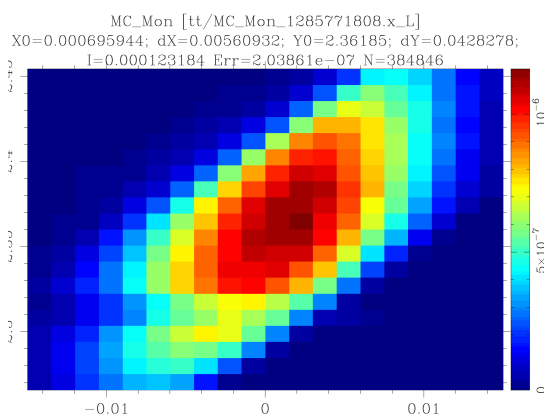


Figure 11c. The *Example 7* with a thick crystal monochromator. The upper border of the spot is blurred from crystal thickness and multiple scattering. The wavelength distribution shows a dip at 2.32 Å from higher order scattering.

5.3 Samples

In this section we shall present the most common material types to be included in instrument models. As explained in section 3, scattering materials are an essential part of virtual experiments. In real experiments, materials are primarily used as samples, and also contribute as crystal monochromators or beam filters (graphite, sapphire, beryllium). But they also give a parasitic signal appearing as a '*background*' (that is all we do not want or do not understand) for instance in sample environment (cryostat, furnace, sample containers) or radiological shielding (concrete, polyethylene, boronated materials, wood, cadmium, lead). The neutron scattering from all these materials, where ever they are positioned in the model, may be simulated with *McStas*.

Neutron scattering may be categorized into two types of processes. Schematically, *coherent* scattering takes place when neutrons encounter ordered matter. This is the case mostly on the material atomic/molecular structure (crystalline, liquid or gas) where atoms are arranged with preferred distances d that neutrons will interact with, following the Bragg law seen in section 5.2.5. This type of interaction is labelled as elastic, as the neutron does not exchange energy but only momentum, that is change its direction but not its velocity. An energy transfer may also be considered when the material either gives or pumps energy to/from the neutron. This energy corresponds to propagating waves in the atomic/molecular structure, such as phonons (sound waves) or magnons (spin waves), or coherent fluctuations of the structure when reaching material phase transitions. All these cases correspond with inelastic coherent scattering, which usually shows quite sharp features in momentum and energy transfer. The *incoherent* scattering corresponds to interaction with disordered matter, that is for instance impurities and isotopic composition, crystal twinning and boundaries, Brownian motion and heat diffusion. Such interactions often appear as broad momentum and energy transfer contributions.

The details of the scattering laws which apply in samples are beyond the scope of this paper, and we re-direct the reader to reference books [14] for more information.

5.3.1 Powder structures

Among the most common materials measured with neutron scattering techniques are powders, which are composed of a large number of tiny single crystals. The scattering intensity, by means of constructive interference and averaging over all crystallites, appears as well defined rings around the out-going direct beam direction. These rings are also present in liquids, but are much smoother, as we shall see in next section.

Let us consider a material of density ρ made of N scattering units each with a unit cell volume V_0 arranged with lattice spacings $d_Q=2\pi/Q$ associated to structure factors $F(Q)$ with equivalent reflection multiplicities j_Q . These structure factors characterize the efficiency of the reflection with momentum exchange Q . Following Squires [14] the scattering probability for an incoming neutron with wavelength λ penetrating along a distance x into the material is about $1-\exp(-\rho\sigma_{cone} x)$ where σ_{cone} is the so-called coherent elastic cross-section of the ring,

$$\sigma_{cone} = \frac{N \pi \lambda^2 j_Q |F(Q)|^2}{V_0 Q} .$$

This relation is only valid under certain conditions, among which $d_Q > \lambda/2$, from the Bragg law. As the possible lattice spacings d in the material can not exceed a maximum value (for instance the inter-atomic distance), it appears that as the neutron wavelength increases, the number of visible rings in the diffractogram will decrease, until no more scattering is possible above the so-called *Bragg edge*, where materials become transparent to neutrons (except for absorption and incoherent scattering). This is why most of the diffractometers use thermal and hot neutrons. Cold neutrons can only scatter on large distance arrangements in materials, *e.g.* in larger molecules and proteins.


```

/*****
* Instrument: powder_simple.instr
* %Description
* A powder scattering example.
*
* %Parameters
* lambda: [Angs] incoming neutron wavelength (monochromatic)
* material: [Angs] Powder structure file (lazy/fullprof/crystallographica)
*****/
DEFINE INSTRUMENT powder_simple (lambda=2.36, string material="Na2Ca3Al2F14.laz")
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.01, xw=0.01, yh=0.01,
                                Lambda0=lambda, dLambda=lambda*0.01)
    AT (0,0,0) ABSOLUTE
COMPONENT powder= PowderN(reflections = material, radius = 0.005, yheight = 0.05)
    AT (0,0,0.5) RELATIVE PREVIOUS
COMPONENT banana= Monitor_nd(xwidth=1, yheight=0.3,
                             options="banana ; theta limits=[10,130] bins=240 ; y bins=50")
    AT (0,0,0) RELATIVE PREVIOUS
END

```

Example 8. A simple powder diffraction example.

The equations from [14] relating to powders have been directly implemented in the *PowderN* component [15]. This handles single, coherent scattering and many *d*-spacing structure factors, with absorption correction and incoherent elastic scattering. However, no multiple or inelastic scattering is taken into account, which the *Isotropic_Sqw* component can cope with, in its powder mode (see section 5.3.2 below).

In the *Example 8*, we present a usage example which produces so-called Debye-Scherrer rings from a structure factor list. The model geometry is shown in *Figure 12a*, and the 2D ideal diffraction detector in *Figure 12b* for a $\text{Na}_2\text{Ca}_3\text{Al}_2\text{F}_{14}$ reference powder. The choice of the material may be any file adapted from *Lazy/Pulverix* implemented in the ICSD database [16] and *Fullprof* [17] (extension *.laz*) or *Crystallographica* (extension *.lau* also used for crystals). Currently, *McStas* includes a material data base of about 60 powder and crystal definitions commonly used in neutron scattering. These can be listed from the *McGUI/Help/Component Library Index* menu item, and you may easily add your own materials. The material volume may be a box, a sphere and a cylinder, which all can be bulk or hollow geometries, including concentric arrangements, as we shall see in section 7.2.

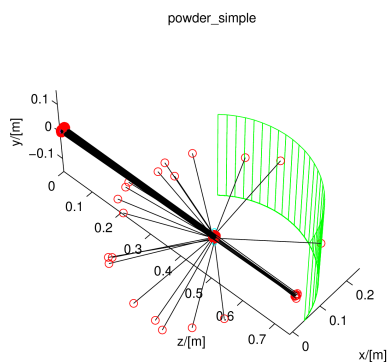


Figure 12a. Geometry of the *Example 8* simple powder diffraction setup, showing a few diffracted neutron trajectories. The beam is attenuated and scattered in the sample.

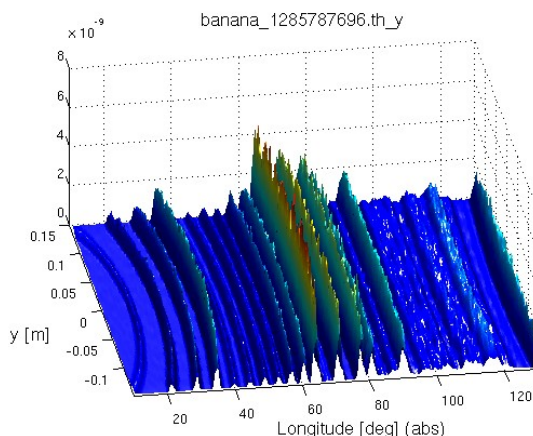


Figure 12b. Diffraction pattern obtained from a $\text{Na}_2\text{Ca}_3\text{Al}_2\text{F}_{14}$ powder at $\lambda=2.36 \text{ \AA}$, showing intensity vs. horizontal angle vs. vertical position along the detector.

5.3.2 Liquid, powder and gas elastic and inelastic scattering

There is no periodic arrangement in liquids, as atoms and molecules move continuously. However, there are still preferred distances, for instance the one between two atoms, seen as hard spheres that would slip one on the other, but can not inter-penetrate, or the various coordination sphere radii. The Bragg law still applies on such distances, and will result in somewhat blurred Debye-Scherrer rings compared with powders. In the following, we focus on liquid

materials, but the same formalism can be transposed from liquids to powders (where structural lines are sharper) and gas.

Following the literature again [14,18,19], we find out that the total scattering cross-section reads, with usual notations,

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

where E_i is the incident neutron energy, ω and q are the energy and momentum exchange between the material and the neutron, σ is the material bound cross-section, $d\Omega$ is a solid angle, and $S(q, \omega)$ is the dynamic structure factor which is the double Fourier transform of the spatial and time atomic position distributions in the material. This quantity is characteristic of each material, and exhibits for instance the structure factors $|F(Q)|$ introduced in the powder section **5.3.1** above as peaks along the $\omega=0$ elastic line. The inelastic spectrum $\omega \neq 0$ shows dispersion curves such as phonons and broad quasi-elastic lines. In liquids, the structural information is fuzzy, and phonon lines are broad.

The *Isotropic_Sqw* component models any isotropic density material, such as liquids, polymers and amorphous materials, gas and powders. It handles coherent and incoherent scattering both with elastic and inelastic processes. Multiple scattering and secondary extinction are taken into account. The material volume may be a box, a sphere and a cylinder, which all can be bulk or hollow geometries, including concentric arrangements. An additional complex geometry enables to use any point set to describe the material volume (*geomview* OFF file). The material is specified *e.g.* with a data file, which is either a structure file such as for the *PowderN* component (extension *.laz* or *.lau*), or a $S(q)$ liquid structure factor file (extension *.qSq*), in which case no inelastic scattering is then modelled. In order to model the dynamical part, $S(q, \omega)$ **tables** must be provided (extension *.sqw*), and prepared for instance from molecular dynamics or experiments as explained in section **8.2** below. Powders are handled less accurately in *Isotropic_Sqw* than with the *PowderN* component. Gas dynamical structure factors may be approximated as a liquid with very low density.

The *Example 9* lists an instrument model that makes use of this component, in the spirit of the powder *Example 8*. The model geometry is similar to the one shown in *Figure 12a*. Running this example is somewhat longer than the previous ones, due to the fact that the sample model is much more complex and includes multiple scattering. The simulation results are shown in *Figure 13*, when using as material information the dynamic structure factor $S(q, \omega)$ prepared from a liquid rubidium, classical, molecular dynamics simulation [19].

The liquid structure for a liquid rubidium sample can be analysed from a diffraction pattern, as shown in *Figure 13* (left), which may actually be obtained on real diffractometers. The arcs are portions of Debye-Scherrer rings in a liquids, and their radial integration provides the $S(q)$ structure factor, which characterizes the atomic distances in the liquid.

The angle-energy spectrum is shown in *Figure 13* (right). The elastic line is the horizontal one around the neutron incident energy of about 14 meV. The more intense peaks around 30, 60 and 100 degrees angles correspond with the $S(q)$ structural information of the diffraction pattern on the right side. Faint arches can be seen to converge to these peaks, scattered by phonons in the liquid. This latter plot can not be measured directly with current detectors, as we have used a neutron energy sensitive monitor. With real instruments, an angle/time distribution would provide a similar, but indirect, result when using *e.g.* a time-of-flight spectrometer.

Such a model is already quite realistic, even though no proper primary spectrometer is actually modelled here to provide a monochromatic beam. This demonstrates the essential role of sample kernels in virtual experiments, which should be complemented with an accurate description of the neutron optics which compose the instrument. This will be achieved in section **6.1**.

```

/*****
* Instrument: liquid_simple.instr
* %Description
* A liquid scattering example.
*
* %Parameters
* lambda: [Angs] incoming neutron wavelength (monochromatic)
* material: [Angs] liquid or powder description file (lazy/fullprof/crystallographica/qSq/Sqw)
*****/
DEFINE INSTRUMENT liquid_simple (lambda=2.36, string coh="Rb_liq_coh.sqw", string
inc="Rb_liq_inc.sqw")
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.01, xw=0.01, yh=0.01,
Lambda0=lambda, dLambda=lambda*0.01)
AT (0,0,0) ABSOLUTE
COMPONENT sample= Isotropic_Sqw(Sqw_coh = coh, Sqw_inc = inc, radius = 0.005, yheight = 0.05)
AT (0,0,0.5) RELATIVE PREVIOUS
COMPONENT banana= Monitor_nD(xwidth=1, yheight=0.3,
options="banana ; abs theta limits=[10,130] bins=240 ; y bins=50")
AT (0,0,0) RELATIVE PREVIOUS
COMPONENT bananE= Monitor_nD(xwidth=1, yheight=0.3,
options="banana ; angle limits=[10,130] bins=120 ; auto energy bins=100")
AT (0,0,0) RELATIVE PREVIOUS
END

```

Example 9. A simple liquid scattering example, which also works with powders and gases.

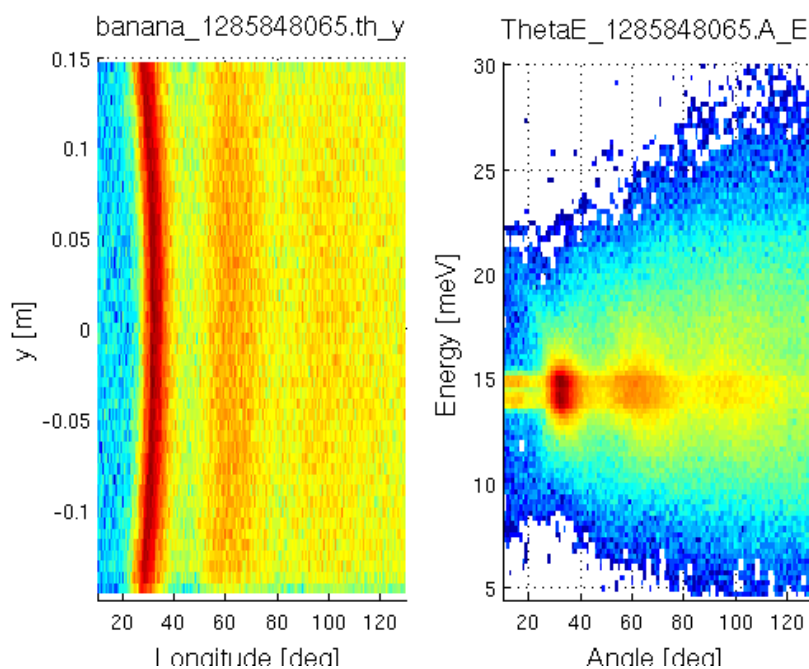


Figure 13. Neutron scattering from a liquid rubidium sample, obtained from the *Example 9* at $\lambda=2.36$ Å. The left plot shows the liquid Debye-Scherrer rings as obtained from a diffractometer 2D detector, whereas the right one shows a radial angle-energy signal with inelastic contribution. Intensity is shown in log scale, with colors ranging from blue (low) to red (high).

5.3.3 Single crystal structures

When the sample is a single crystal, the averaging on many crystallites that is responsible for the scattering rings in a powder does not apply. The Bragg law is still valid, but similarly as a mirror, each atomic plane selects a single reflected monochromatic ray. As there are many structural planes available, a polychromatic neutron beam will be scattered as a large number of distinct rays, forming spots on detectors. This happens in the monochromators discussed in section 5.2.5, but in this case only one reflection of interest is used, the others are scattered around, generating background.

The scattering cross-section from a set of reflections $\vec{\tau}$ follows a similar law as the one used for powders in section 5.3.1. However, this time the momenta exchange are vectors instead of just norms. Then the coherent elastic cross section is

$$\sigma(\vec{k}_i) = \int_{4\pi} \sum_{\vec{\tau}} \frac{N(2\pi)^3}{V_0} \delta(\vec{\tau} - (\vec{k}_i - \vec{k}_f)) |F(\vec{\tau})|^2 d\Omega ,$$

which simply indicates that the beam is only scattered by the reflections which satisfy the momentum conservation $\vec{\tau} = \vec{k}_i - \vec{k}_f$ that is the Bragg law (again), on all possible outgoing directions.

```

/*****
* Instrument: single_crystal.instr
* %Description
* A single crystal scattering example.
*
* %Parameters
* lambda: [Angs] neutron wavelength selected by the monochromator
*****/
DEFINE INSTRUMENT single_crystal (lambda=1, string material="C_graphite.lau")

TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.01, xw=0.01, yh=0.01,
                                Lambda0=lambda dLambda=0.2*lambda)

  AT (0,0,0) ABSOLUTE
COMPONENT SX = Single_crystal(
  xwidth = 0.002, yheight = 0.1, zthick = 0.1,
  mosaic = 30, reflections = material, barns=0,
  ax = 0, ay = 2.14, az = -1.24,
  bx = 0, by = 0, bz = 2.47,
  cx = 6.71, cy = 0, cz = 0,
  absorption = 0.014, incoherent = 0.004)
  AT (0,0,0.5) RELATIVE PREVIOUS ROTATED (0,45,0) RELATIVE PREVIOUS
COMPONENT banana= Monitor_nd(xwidth=1, yheight=1,
                             options="banana ; theta limits=[10,130] bins=240 ; y bins=100")

  AT (0,0,0.5) RELATIVE Source
END

```

Example10. A single crystal scattering example, with a large banana shaped detector. The graphite crystal is tilted by 45°.

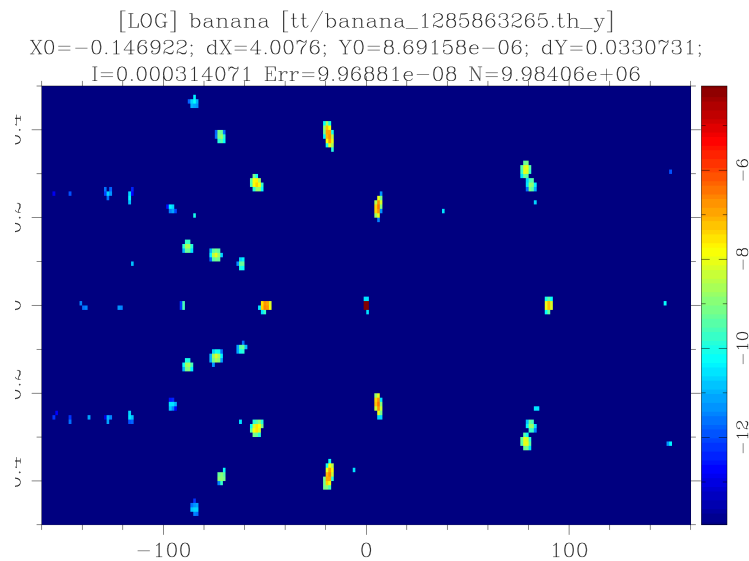


Figure 14. Neutron scattering from a single crystal of graphite from a neutron beam around $\lambda=1$ Å, obtained from the *Example 10*. Intensity is shown as a function of the horizontal angle and vertical coordinate, in log scale, with colors ranging from blue (low) to red (high).

The *Single_crystal* component is used the same way as the *PowderN*, but only accepts *.lau* type files from *Crystallographica* [20]. This component models coherent and incoherent elastic scattering, with multiple scattering and secondary absorption. The material volume may be a box, a sphere and a cylinder, which all can be bulk or hollow geometries, including concentric arrangements. An additional complex geometry enables to use any point set to describe the material volume (*geomview* OFF file). In order to demonstrate its usage, we re-use the *Examples* 6 and 7 with a wider incoming wavelength range, and add a larger detector, as shown in *Example* 10. The instrument geometry resembles the *Figure* 12a, with a tilted graphite plate at the sample. As expected, the scattering shows a number of spots, which each select a single wavelength. The central spot is the direct, transmitted beam.

Currently, McStas does not provide simple ways to add inelastic scattering on top of a mono-crystalline structure, even though there is a way to simulate the neutron scattering on a simple phonon dispersion.

5.3.4 Other sample models

A number of other sample types may be modelled with *McStas*, and we shall thus present some possibilities, without actually demonstrating their usage in the frame of this paper. We point the reader to the *McStas Component Manual* [9] and the *Library Index* for more details.

The sample models such as *PowderN*, *Isotropic_Sqw* and *Single_crystal* may be used for **small angle scattering**, as long as the data files do include large scale structure data. However, no special form factors will then be taken into account. The *Sans_sphere* component may be used to model dilute colloids of hard spheres, the *SANS_Guinier* models a scattering unit with a gyration radius, and the *SANS_DebyeS* models a simple highly ordered diblock copolymer.

As we have already mentioned, the *Isotropic_Sqw* component may model inelastic scattering for powders and liquids. But the *Single_crystal* component does not implement inelastic scattering from crystals. A way to overcome this is to use the **Phonon_simple** component, which handles a single dispersion relation in cubic crystals.

Last, we mention dedicated samples for **reflectometry**. The simplest *SiC* component models a single bulk plate, and the more advanced *Multilayer_sample* can be used to model any layered arrangement.

5.4 Detectors

Most of the detector models used in the *McStas* instrument descriptions are ideal, and they record events in space and time, as this is the case for real devices, but also monitor a wide range of other beam parameters (energy, divergence, spin, ...).

There is a large list of so-called monitor components, which usually require to specify some histogram (tally) recording bounds and binning. A more unified specification of these monitoring parameters resides in the **Monitor_nD** component, in which an option parameter specifies explicitly what to record, and the monitor geometry. All of the previously shown *Examples* make use of this component, often with the option="... auto ..." parameter to determine the best histogram bounds from incoming neutron events.

An other class of monitors are the event type ones. These monitors do not create histograms but just flush neutron events into a file, for further analysis. Such feature may be obtained from the *Monitor_nD* with the *options="source"* parameter or the *Virtual_output* component.

A much more advanced model will be introduced in section 7.3 with the *PSD_Detector* gas detector component.

6. VIRTUAL EXPERIMENT SIMULATION EXAMPLES

We shall now use the items presented in the section 5 (sources, neutron optics, samples, detectors) to describe simple, but realistic, virtual experiment models. For more advanced components and usage examples, refer to section 7. Running these models will be slightly longer than those demonstrated in section 5, lasting a few minutes on current computers.

6.1 A time of flight spectrometer for liquids

The idea is here to model a disk chopper time-of-flight spectrometer with a liquid sample and a large banana shaped detector. This consists essentially in assembling the disk chopper *Example 3* (section 5.2.3) and the isotropic sample *Example 9* (section 5.3.2), as shown in the *Example 11*. The model geometry is shown in *Figure 15a*. The SPLIT keyword at the sample position defines a partitioned sampling mechanism to be applied, in order to greatly improve the simulation efficiency (see section 2.3).

The instrument detector image result is shown in *Figure 15b*, and resembles in many ways *Figure 13*, but with a time axis in place of the energy axis. This distribution can be directly measured on existing time-of-flight spectrometers. As a standard data analysis procedure, the time axis is usually processed and converted into the energy transfer ω , whereas the angle axis provides the momentum transfer q , so that it actually provides the sample dynamic structure factor $S(q, \omega)$.

```

/*****
* Instrument: liquid_spectrometer.instr
* %Description
* A disk chopper spectrometer with liquid sample and large detector.
*
* %Parameters
* lambda: [Angs] neutron wavelength selected by the chopper system
* frequency: [Hz] chopper rotation frequency, omega=2*PI*frequency, rpm=60*frequency
* d_cc: [m] distance between the two choppers. Chopper opening slits is 10 deg here.
* coh: [str] liquid or powder coherent scattering (Sqw or powder format)
* inc: [str] liquid or powder incoherent scattering (Sqw or NULL for isotropic elastic)
*****/
DEFINE INSTRUMENT liquid_spectrometer (lambda=4, frequency=200, d_cc=5,
string coh="Rb_liq_coh.sqw", string inc="Rb_liq_inc.sqw")
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.03, xw=0.03, yh=0.05,
Lambda0=lambda, dLambda=lambda*0.01)
AT (0,0,0) ABSOLUTE
COMPONENT Chop1 = DiskChopper(omega=2*PI*frequency, R=0.3, h=0.05, theta_0=10, n=1, IsFirst=1)
AT (0, 0, 1) RELATIVE PREVIOUS
COMPONENT guide = Guide(w1=0.03, h1=0.05, w2=0.03, h2=0.05, l=d_cc-0.1)
AT (0, 0, 0.05) RELATIVE PREVIOUS
COMPONENT Chop2 = COPY(Chop1)(IsFirst=0, phi_0=2*PI*frequency*d_cc*lambda/3956*RAD2DEG)
AT (0, 0, d_cc) RELATIVE Chop1
COMPONENT CG_Mon= Monitor_nd(xwidth=0.03, yheight=0.05, options="auto time")
AT (0,0,0.1) RELATIVE PREVIOUS

SPLIT COMPONENT sample= Isotropic_Sqw(Sqw_coh = coh, Sqw_inc = inc, radius = 0.005, yheight =
0.05)
AT (0,0,0.5) RELATIVE PREVIOUS
COMPONENT detect= Monitor_nd(xwidth=2.5, yheight=1,
options="banana ; auto time bins=256 ; angle limits=[10,130] bins=120")
AT (0,0,0) RELATIVE PREVIOUS
END

```

Example 11. A liquid scattering time-of-flight spectrometer virtual experiment. This simulation requires a longer computing time to produce the *Figure 15b*.

Of course, in this virtual experiment, a similar information has been used from the coherent and incoherent parameter files. But these, obtained here from a molecular dynamics simulation (see section 5.3.2 and [19]) do not include any instrumental effects. In practice, no instrument is perfect, so that the measured dynamical structure factor will not match exactly the simulated one. The differences are related to the imperfections which hinder any measurement. In the case of this liquid spectrometer, virtual experiment, we may notice a number of differences. The most striking one is the parallax effect which is not corrected here, due to the fact that for a cylindrical detector, the time of flight from the sample to the detector is shorter in the equatorial plane, and longer when hitting the top and bottom detector sides. This results in a broadening of the time distribution, which can easily be corrected in practice. Also, the neutron beam reaching the sample position suffers a number of artefacts. In particular, the neutron energy and pulse length defined by the choppers are not perfect, so that it also broadens the time axis on the detector. And, as the neutron beam is not strictly parallel, a distribution of incoming neutron momentum directions will result in a broadening of the angular axis. Finally, a variation of the time of flight originates from the sample volume itself which introduces an additional path length to travel to the detector. All of these effects are common on all neutron scattering instruments and may be analysed and corrected. But the addition of all imperfections may not be easily corrected, as they are usually cross-correlated. The more complex the instrument is, the more measurement artefacts appear, and the harder it is to apply

simple analytic corrections.

Besides the instrument parasitic effects, the incoherent and multiple scattering can hardly be removed from experiments, where they often partly hide the coherent scattering contribution. In the *Figure 15b*, these are mainly visible at low scattering angle, around the elastic line. We shall present a method in section 7.2 to estimate some of the contributions which contribute in an experimental measurement.

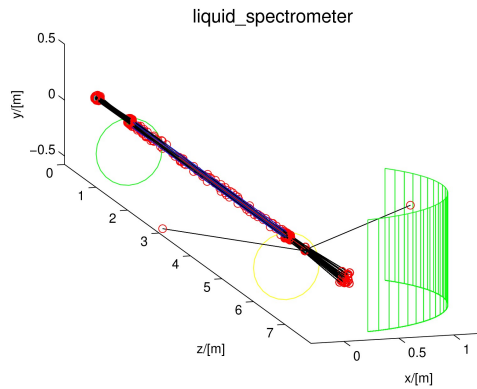


Figure 15a. A disk chopper liquid spectrometer geometry obtained from *Example 11*.

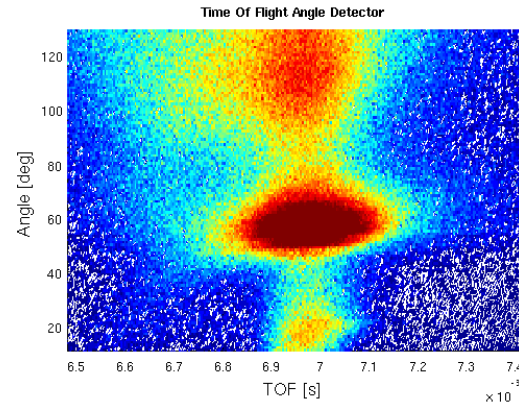


Figure 15b. the radial angle/time-of-flight signal obtained with a liquid rubidium sample, for $\lambda=4 \text{ \AA}$. The vertical central line corresponds to the elastic scattering.

6.2 A powder diffractometer

One of the most widely used neutron instrument is the powder diffractometer. It mainly consists of appending the crystal monochromator *Example 6* (section 5.2.5) and the powder sample *Example 8* (section 5.3.1). However, most of the diffractometers make use of an additional radial collimator such as the *Example 2* (section 5.2.2) in between the sample and the banana shaped detector, in order to improve the angular resolution of the instrument. The resulting powder diffraction instrument description is listed as *Example 12*, and its geometry is shown in *Figure 16*.

This example makes use of the partitioned sampling variance reduction technique (SPLIT keyword, see section 2.3 and 6.1). Also, we are using the monochromator in a double focusing geometry to increase the neutron flux on the sample, by setting the curvature radius from optical formulas. In the instrument description, we are using two detectors which are symmetric in angle in order to double the simulation statistics. However, with a graphite monochromator selecting 1.4 \AA neutrons, the transmitted beam from the monochromator intersects the symmetric detector area (on the left side of *Figure 16*), generating an intense parasitic peak. A way to remove this artefact is to get rid of the transmitted beam by means of a *McStas* grammar feature which enables to customize the behaviour of component instances using the EXTEND keyword. This latter is applied after the monochromator instance.

```

/*****
* Instrument: powder_diffraction.instr
* %Description
* A powder diffractometer with powder sample, radial collimator and large detector.
*
* %Parameters
* lambda: [Angs] neutron wavelength selected by the crystal monochromator
* DM: [Angs] Monochromator d-spacing, default is Graphite
* material: [Angs] powder structure file (lazy/fullprof/crystallographica)
* ALPHA: [arc min] collimator divergence. One arc minute is a degree
*****/
DEFINE INSTRUMENT powder_diffraction (lambda=1.4, DM=3.355, ALPHA=10,
string material="Na2Ca3Al2F14.laz" )
DECLARE %{
double theta; /* half take-off, used further in the instrument TRACE */
%}
INITIALIZE %{
theta = asin(lambda/DM/2)*RAD2DEG; /* also called A1 on triple-axis machines */
printf("%s: 2*theta=%g [deg] take-off angle\n", NAME_CURRENT_COMP, 2*theta);
%}
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.1, xw=0.1, yh=0.1,
Lambda0=lambda, dLambda=lambda*0.04)
AT (0,0,0) ABSOLUTE
COMPONENT MC = Monochromator_curved(width = 0.10, height = 0.10, DM=DM,
RV=2*sin(theta*DEG2RAD), RH=2/sin(theta*DEG2RAD))
AT (0,0,1) RELATIVE PREVIOUS ROTATED (0,theta,0) RELATIVE PREVIOUS
EXTEND %{
if (!SCATTERED) ABSORB; /* absorb non scattered events = direct beam */
%}
COMPONENT MC_out= Arm() AT (0,0,0) RELATIVE PREVIOUS ROTATED (0,theta,0) RELATIVE PREVIOUS
SPLIT COMPONENT sample= PowderN(reflections = material, radius = 0.005, yheight = 0.05 ,
d_phi=atan2(0.3,1)*RAD2DEG)
AT (0,0,1) RELATIVE PREVIOUS
COMPONENT radial = Collimator_radial(
radius=0.6, yheight = 0.3, length = 0.30, divergence = ALPHA,
theta_min=5, theta_max=140)
AT (0,0,0) RELATIVE PREVIOUS
COMPONENT psd = Monitor_nD(xwidth=2, yheight=0.3,
options="banana ; abs theta limits=[5,140] bins=270 ; y")
AT (0,0,0) RELATIVE PREVIOUS
COMPONENT banana = Monitor_nD(xwidth=2.01, yheight=0.3,
options="banana ; angle limits=[5,140] bins=270")
AT (0,0,0) RELATIVE PREVIOUS
END

```

Example 12. A powder diffractometer virtual experiment. This simulation requires a longer computing time to produce the *Figure 17*.

The results from the powder diffractometer simulation are shown in *Figure 17*, as two diffractograms. The left one corresponds to the position sensitive detector image where Debye-Scherrer ring arcs are seen (see section 5.3.1). The radial integration of these rings produces the second plot on the right side, which is a typical powder diffraction pattern, which implicitly includes instrumental resolution function. In practice, using a curved monochromator results in a broadening of the diffraction rings out of the equatorial scattering plane.

Replacing the *PowderN* instance with an *Isotropic_Sqw* one would even bring an estimate of the multiple scattering, as well as support for liquids, possibly with inelastic background when given proper $S(q, \omega)$ data.

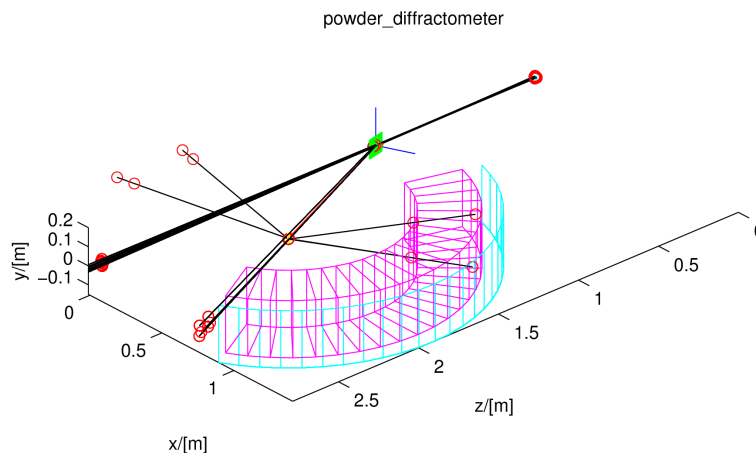


Figure 16. A powder diffractometer geometry obtained from the *Example 12* for an incoming neutron wavelength of 1.4 Å.

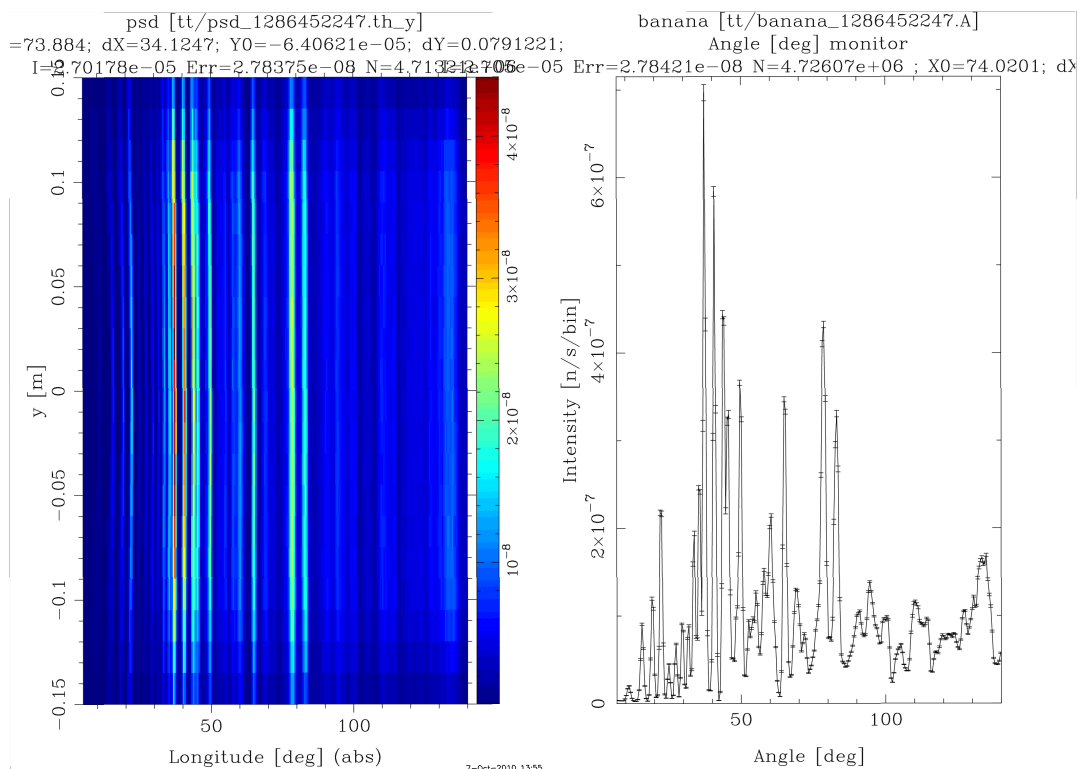


Figure 17. The Na₂Ca₃Al₂F₁₄ powder diffractogram simulated with the *Example 12* obtained at $\lambda=1.4$ Å. Left side shows the angle-height diffractogram with Debye-Scherrer arcs, and right side shows the radially integrated angle diffractogram.

7. ADVANCED NEUTRON OPTICS SIMULATION EXAMPLES

It is possible to use the previous examples in order to estimate other effects such as a background level from an instrument part. The models presented in this section are certainly the most advanced presented in this work, and their execution requires a longer simulation time, up to 15 minutes on a current desktop computer. However, the corresponding descriptions remain very compact, and focus on non-trivial effects which do arise when performing

neutron scattering experiments.

7.1 Monochromators thickness and background sources

The use of heavy shielding around monochromator housings can be understood, as a large portion of the incoming beam scatters in all directions, and the following neutron instrument usually only collects one (see *Figure 14* in section 5.3.3). The remaining scattered beams generate additional scattering when hitting again other materials. In the end, a significant amount of background may arise. In order to get an insight into this effect, we shall start from the monochromator *Example 7* with a single crystal model (section 5.2), and add an aluminium mount aside, as crystal blade holder model. In order for the scattered beam on the crystal to eventually be scattered again on the mount, we copy the bar instance after the single crystal. The model geometry is very similar to the *Figure 11a*, but with an additional pillar beside the diffractometer crystal.

```

/*****
* Instrument: monochromator_background.instr
* %Description
* A monochromator example.
*
* %Parameters
* lambda:      [Angs] neutron wavelength selected by the monochromator
* DM:         [Angs] Monochromator d-spacing, default is 002 Graphite
*****/
DEFINE INSTRUMENT monochromator_background (lambda=2.36, DM=3.355)
DECLARE %{
  double theta;      /* half take-off, used further in the instrument TRACE */
  double flag_holder; /* flag which is set when a scattering event occurs on the crystal pillar */
%}
INITIALIZE %{
  theta = asin(lambda/DM/2)*RAD2DEG; /* also called A1 on triple-axis machines */
  printf("%s: 2*theta=%g [deg] take-off angle\n", NAME_CURRENT_COMP, 2*theta);
%}

TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.05, xw=0.1, yh=0.1,
                                Lambda0=lambda, dLambda=lambda*0.4)
  AT (0,0,0) ABSOLUTE /* large incoming beam */
COMPONENT MC_in = Progress_bar()
  AT (0,0,0.5) RELATIVE PREVIOUS ROTATED (0,theta,0) RELATIVE PREVIOUS
  EXTEND %{
    flag_holder=0; /* neutron events entering here have not scattered yet */
  %}
COMPONENT HoldIn= PowderN(reflections="Al.laz", xwidth = 0.01, zthick = 0.01, yheight = 0.1 )
  AT (0,0,-0.055) RELATIVE MC_in /* this is the pillar */
  EXTEND %{
    flag_holder=SCATTERED; /* set to true when scattering events happen in the pillar */
  %}
COMPONENT SX = Single_crystal(
  xwidth = 0.002, yheight = 0.1, zthick = 0.1,
  mosaic = 30, reflections = "C_graphite.lau", barns=0,
  ax = 0, ay = 2.14, az = -1.24,
  bx = 0, by = 0, bz = 2.47,
  cx = 6.71, cy = 0, cz = 0,
  absorption = 0.014, incoherent = 0.004)
  AT (0,0,0) RELATIVE MC_in
COMPONENT HoldOut = COPY(HoldIn) /* the same pillar but for outgoing neutrons */
  AT (0,0,-0.055) RELATIVE MC_in
  EXTEND %{
    if (SCATTERED) flag_holder=1; /* set to true when scattering events happen in the pillar */
  %}
COMPONENT MC_crystal= Monitor_nD(xwidth=1, yheight=1,
  options="sphere ; theta limits=[-180 180]; y", bins=180)
  AT (0,0,0.5) RELATIVE Source
COMPONENT MC_holder = COPY(MC_crystal) WHEN(flag_holder)
  AT (0,0,0.5) RELATIVE Source
END

```

Example 13. An advanced monochromator setup, which estimates the background generated from the crystal mount.

In this model, we use the EXTEND keyword which enables to customize the behaviour of instrument components. In this case, we set a flag when a scattering event occurs in the aluminium pillar. This flag is then used to monitor both the

signal from the monochromator crystal blade and the pillar, and the part that has scattered in the aluminium bar only. The beam image is recorded on a 1 m diameter sphere centred on the monochromator position, with its angle centred on the direct transmitted beam.

The resulting total scattering is recorded and shown in *Figure 20*. The graphite crystal scatters the (002) reflection around 41.2° take-off angle on the left plot, corresponding to a wavelength $\lambda=2.36 \text{ \AA}$. The strongest spot in the centre of the plot is the transmitted (and attenuated) beam through the crystal. A large number of weaker diffraction spots are scattered, both in and out of the scattering plane. The shape of these spots depends on the crystal thickness and tilt, as well as the beam attenuation from the neighbouring pillar. Below the Laue Bragg scattering from the wide incoming neutron wavelength range appears the pillar powder scattering pattern as a large distribution of Debye-Scherrer rings. The background is essentially backward scattering, and at 90 degrees, centred on the incoming direction. The total additional background from the aluminium structure aside the graphite blade can be estimated to around be 0.7 % of the intensity scattered by the crystal, but may be locally stronger for some angles which correspond to diffraction in aluminium. In the vicinity of the (002) reflection direction at $\lambda=2.36 \text{ \AA}$, this background intensity amounts to about 0.1%.

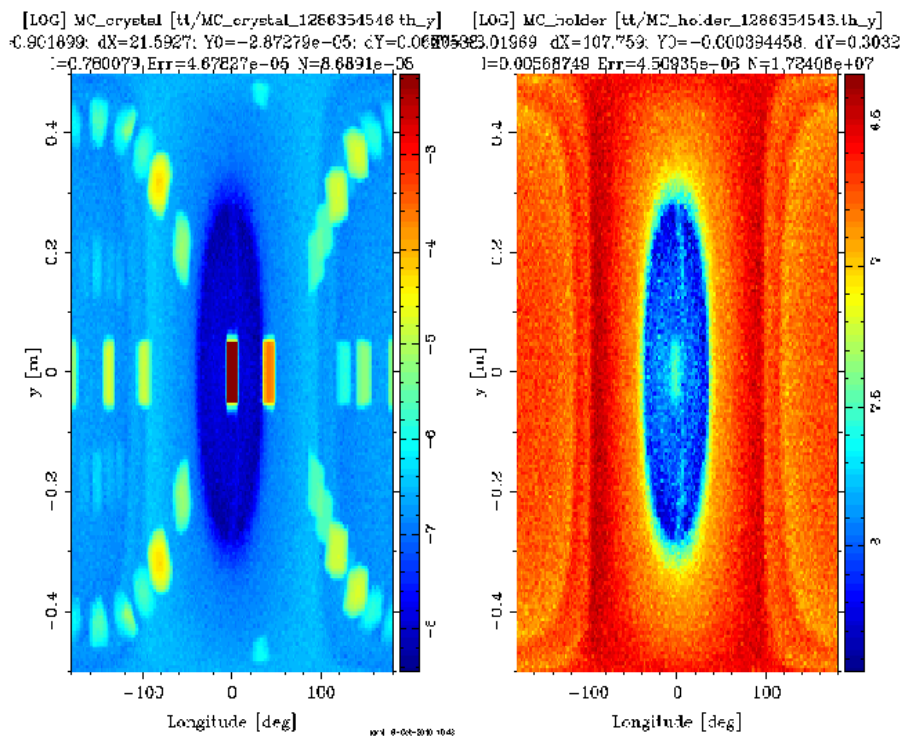


Figure 20. Results from the monochromator *Example 13* around $\lambda=2.36 \text{ \AA}$, showing the total contribution (with crystal diffraction) on the left side, and the pillar contribution on the right side. Intensity is shown in log scale, with colors ranging from blue (low) to red (high).

7.2 Sample environment

An important source of background which appears on acquired scattering signals originates from the sample environments, that is any material close to the sample which also scatters the neutron beam. This particularly holds for furnaces, cryostats, magnets, pressure cells, and sample containers. A way to circumvent this contribution is to make use of radial collimators, which select a small gauge volume around the sample, and remove most of the outer scattering. However, real experiments usually also include acquisitions with an empty container, which is afterwards subtracted from the total signal in order to extract the sample contribution.

```

/*****
* Instrument: liquid_sample_environment.instr
* %Description
* A disk chopper spectrometer with liquid sample and large detector, including a
* sample environment.
*
* %Parameters
* lambda: [Angs] neutron wavelength selected by the chopper system
* frequency: [Hz] chopper rotation frequency, omega=2*PI*frequency, rpm=60*frequency
* d_cc: [m] distance between the two choppers. Chopper opening slits is 10 deg here.
* coh: [str] liquid or powder coherent scattering (Sqw or powder format)
* inc: [str] liquid or powder incoherent scattering (Sqw or NULL for isotropic elastic)
*****/
DEFINE INSTRUMENT liquid_sample_environment (lambda=2.36, frequency=200, d_cc=5,
string coh="Rb_liq_coh.sqw", string inc="Rb_liq_inc.sqw")

DECLARE %{
double flag_sample;
%}
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.03, xw=0.03, yh=0.05,
Lambda0=lambda, dLambda=lambda*0.01)
AT (0,0,0) ABSOLUTE
COMPONENT Chop1 = DiskChopper(omega=2*PI*frequency, R=0.3, h=0.05, theta_0=10, n=1, IsFirst=1)
AT (0, 0, 1) RELATIVE PREVIOUS
COMPONENT guide = Guide(w1=0.03, h1=0.05, w2=0.03, h2=0.05, l=d_cc-0.1)
AT (0, 0, 0.05) RELATIVE PREVIOUS
COMPONENT Chop2 = COPY(Chop1)(IsFirst=0, phi_0=2*PI*frequency*d_cc*lambda/3956*RAD2DEG)
AT (0, 0, d_cc) RELATIVE Chop1

SPLIT COMPONENT sample= Progress_bar() AT (0,0,0.5) RELATIVE PREVIOUS
EXTEND %{
flag_sample=0; /* neutron events reaching the sample area have not scattered yet on it */
%}
COMPONENT Env_In= PowderN(reflections="Al.laz", radius=0.035, radius_i=0.035-0.002, yheight=0.1,
concentric=1, tfrac=0.9)
AT (0,0,0) RELATIVE sample
COMPONENT Con_In= PowderN(reflections="Nb.laz", radius=0.005+50e-6, radius_i=0.005, yheight=0.1,
concentric=1, tfrac=0.9)
AT (0,0,0) RELATIVE sample
COMPONENT liquid= Isotropic_Sqw(Sqw_coh = coh, Sqw_inc = inc, p_interact=0.9,
radius = 0.005, yheight = 0.05)
AT (0,0,0) RELATIVE sample
EXTEND %{
flag_sample=SCATTERED; /* sets flag when scattering occurs in the liquid */
%}
COMPONENT Con_Ou= COPY(Con_In)(concentric=0) AT (0,0,0) RELATIVE sample
COMPONENT Env_Ou= COPY(Env_In)(concentric=0) AT (0,0,0) RELATIVE sample

COMPONENT total= Monitor_nD(xwidth=2.5, yheight=1,
options="banana ; auto time bins=256 ; angle limits=[10,130] bins=120")
AT (0,0,0) RELATIVE sample
COMPONENT detect = COPY(total) WHEN(flag_sample)
AT (0,0,0) RELATIVE sample
END

```

Example 14. A liquid scattering time-of-flight spectrometer virtual experiment with a sample container and outer environment. We could even add a last similar monitor to be used WHEN(flag_sample>1) to record only the multiple scattering part from the liquid sample.

The *Example 14* is assembled from the liquid spectrometer *Example 11* (section 6.1), but adds two outer cylinders made of niobium and aluminium to model the sample container and the first sample environment thermal shield. A few dedicated monitors are added at the end of the instrument description, making use of the WHEN keyword, which activates a component instance depending on a condition. This way we are able to separate contributions from the sample itself, and the sample environment. The concentric geometry implemented in most *McStas* sample components enables to place components within others, to model e.g. sample environments. It consists of repeating a component instance symmetrically w.r.t. the central position (e.g. the sample one).

The model geometry is shown in *Figure 21*. The resulting plots are shown in *Figure 22*. Such results are invaluable in order to understand the meaning of a measurement, and the ratio between the sample signal and the other contributions, often labelled as background. The intense contribution from the sample environment and the container justifies the

installation of radial collimators on most spectrometers.

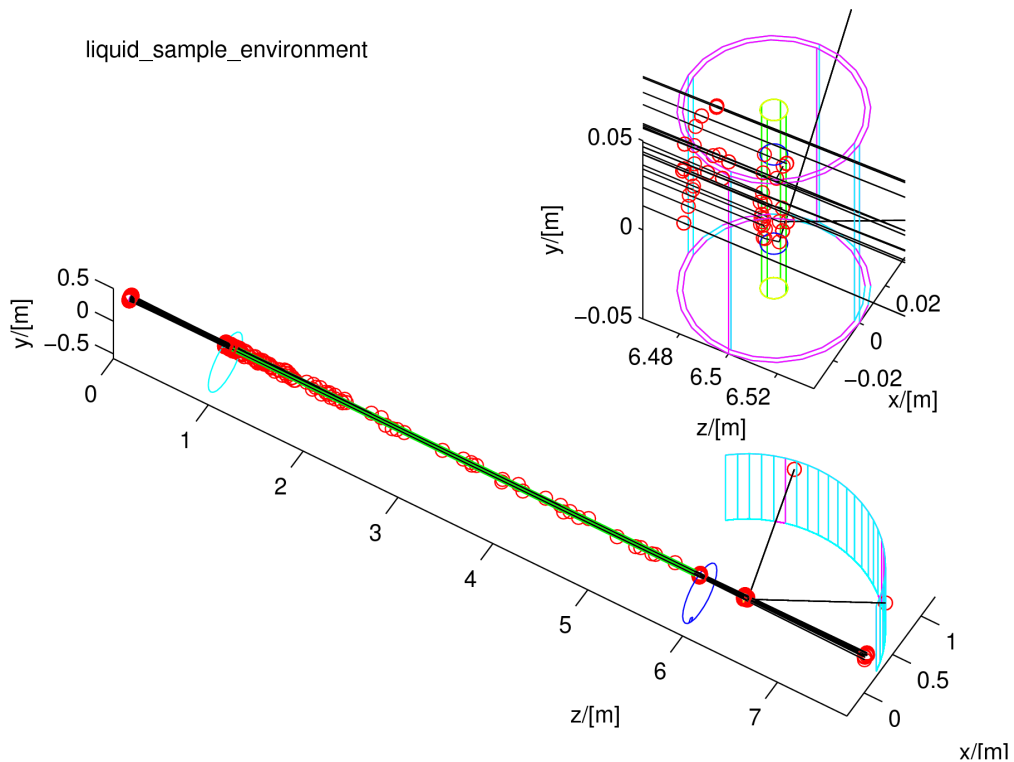


Figure 21. The *Example 14* liquid spectrometer with sample environment geometry (insert). A few neutron trajectories are displayed, with multiple scattering events.

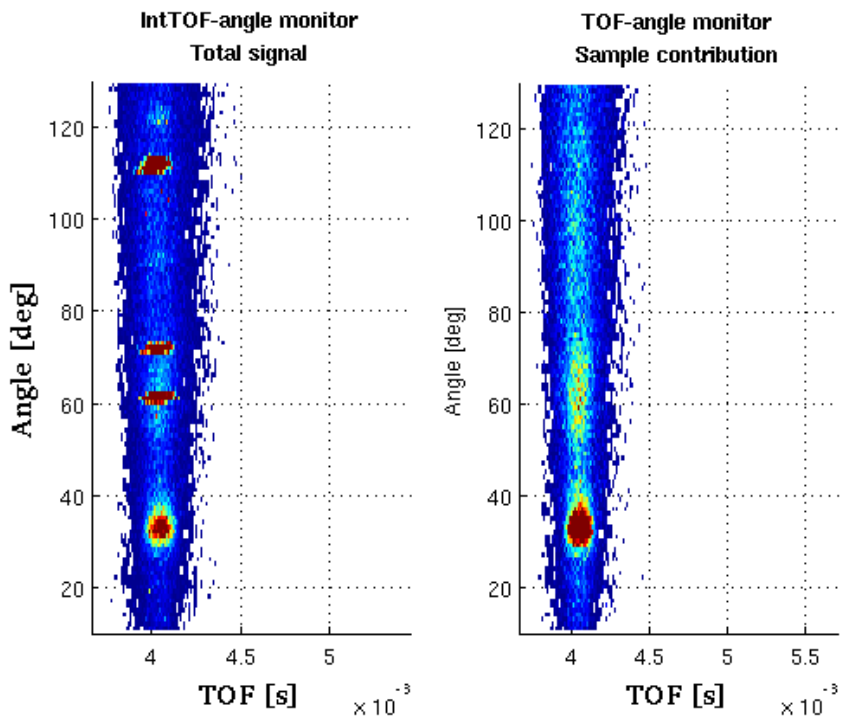


Figure 22. The time-of-flight/angle signal acquired from the liquid spectrometer with sample environment model. The total signal from liquid rubidium in a Nb container and Al external shield is shown on the left, whereas the sample-only contribution is shown on the right. Intensity is shown in log scale, with colors ranging from blue (low) to red (high).

7.4 Gas detectors

Up to now, all the models we have simulated use perfect monitors, as explained in section 5.4. Their efficiency is ideal, and does not depend on the neutron energy. They do not saturate when neutron flux is too high, and they have a perfect spatial response. They are meant to analyse the neutron beam, without introducing additional measurement imperfections.

Most current neutron detectors in the cold-thermal range use a gas technology. Neutrons that enter e.g. a ^3He cell create charges (proton-triton pairs) when absorbed by the atoms. These charges drift in the gas, until stopped and collected by high potential wires.

The *PSD_Detector* has been designed in order to model most of the gas detector physics that takes place in the thermal neutron scattering instruments. The *Example 15* is based on the powder sample *Example 8* (section 5.3.1) which we extend with a 1 mm aluminium window plate followed by a gas volume 3 cm thick, 6 bars cell. We have used a perfect beamstop after the powder sample, which absorbs all non scattered neutrons in an EXTEND block.

```

/*****
* Instrument: powder_detector.instr
* %Description
* A powder scattering example with a realistic gas detector.
*
* %Parameters
* lambda: [Angs] incoming neutron wavelength (monochromatic)
* material: [Angs] Powder structure file (lazy/fullprof/crystallographica)
*****/
DEFINE INSTRUMENT powder_detector (lambda=2.36, string material="Na2Ca3Al2F14.laz")
TRACE
COMPONENT Source= Source_simple(dist=1, radius=0.01, xw=0.01, yh=0.01,
                                Lambda0=lambda, dLambda=lambda*0.01)
    AT (0,0,0) ABSOLUTE
COMPONENT powder= PowderN(reflections = material, radius = 0.005, yheight = 0.05)
    AT (0,0,0.5) RELATIVE PREVIOUS
    EXTEND %{
        if (!SCATTERED) ABSORB;
    %}
COMPONENT Perfect= Monitor_nD(options="x y", bins=512,
                               xwidth=0.26, yheight=0.26, restore_neutron=1)
    AT (0, 0, 0.15) RELATIVE powder
COMPONENT Window = PowderN(
    xwidth=0.26, yheight=0.26, zthick=0.001,
    reflections="Al.laz", tfrac=0.9, frac=0.01)
    AT (0, 0, 0.15) RELATIVE powder
COMPONENT BIDIM26= PSD_Detector(xwidth=0.26, yheight=0.26, depth=0.03,
    FN_Conv="Gas_tables/He3inHe.table", FN_Stop="Gas_tables/He3inCF4.table",
    nx=512, ny=512, filename="BIDIM26.psd", restore_neutron=1)
    AT(0,0,0.03) RELATIVE Window
END

```

Example 15. A simple powder diffraction example which uses a realistic gas detector.

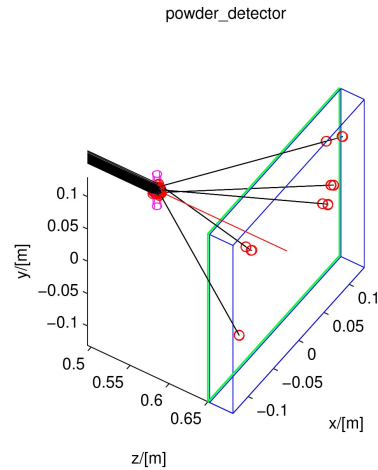


Figure 23. The *Example 15* powder diffractometer with a realistic gas detector model. Scattering events are detected within the gas volume.

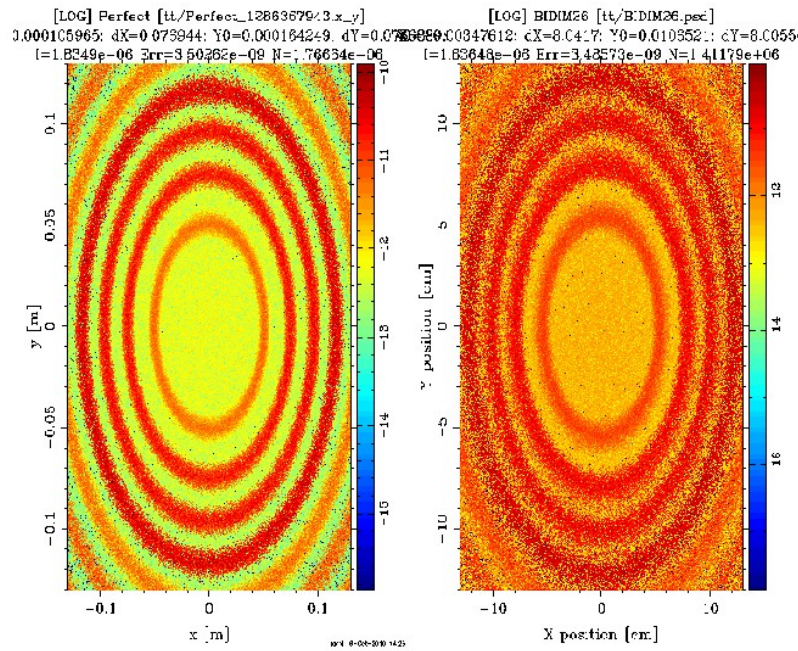


Figure 24. The image recorded from the *Example 15* on an ideal monitor (left) and a gas detector model (right).

The *Example 15* geometry is shown in *Figure 23*, and the powder rings recorded by both a perfect and a more realistic gas detector are shown in *Figure 24*. Comparing the ideal and actually detected image, we notice that the ring diameter is larger in the gas detection than in the ideal monitor (which is just a thin plate). This originates from the fact that events in the gas are detected with an exponentially decreasing efficiency in the cell volume (due to absorption), so that a Debye-Scherrer cone opens when entering the gas. Also, the inner side of the rings is well defined because it corresponds to events detected just when entering the gas cell, whereas the outer ring side is blurred by the penetration is the gas. The resulting contrast on the ideal monitor (left side of *Figure 24*) appears to be, unfortunately, unreachable in reality (right side).

8. COUPLING MCSTAS WITH OTHER TOOLS

The *McStas* software is a open-source, community based package. It has benefited from a large number of user contributions, including some which interface with other simulation packages. Some of the most important interfaces are certainly those which couple *McStas* with the world of reactor and particle physics, and with the world of molecular dynamics to obtain $S(q, \omega)$.

8.1 Reactor/particle physics transport codes

Most reactors that have been built today were designed using MCNP (see section 2.1) and some accelerators have been simulated with e.g. MCNP/X. These codes handle the high energy physics which takes place in fissile materials and occur when high energy particle beams collide with matter. This kind of interaction may generate neutrons, which are then slowed down by e.g. successive Compton scattering processes in moderators. Other codes such as FLUKA (which is closely related to MCNP/X), GEANT4 or TRIPOLI provide the same functionality, and actually all share the wide ENDF material database.

The MCNP type codes can produce neutron event records when passing a surface in the facility model (by means of the PTRAC card) and the *Virtual_mncp_input* McStas component can read these files and send the neutron events into an instrument description, thus replacing the usual source component. Such an advanced source model contains all the neutron parameter correlations, such as any anisotropy from the geometry, whereas the default *McStas* source components (see section 5.1) are approximated by homogeneous event sources.

However, the neutron event files generated by e.g. MCNP or TRIPOLI are usually very large, and contain a finite number of particles, which then limits the achievable accuracy from a *McStas* simulation.

8.2 Obtaining dynamical structure factors

As we have seen in section 5.3, a large set of material structures may be obtained from crystallographic databases. However, no similar knowledge base exists regarding material dynamics. We propose here two procedures in order to obtain such data, in order to simulate inelastic scattering contributions in isotropic materials (see section 5.3.2).

8.2.1 From experiments

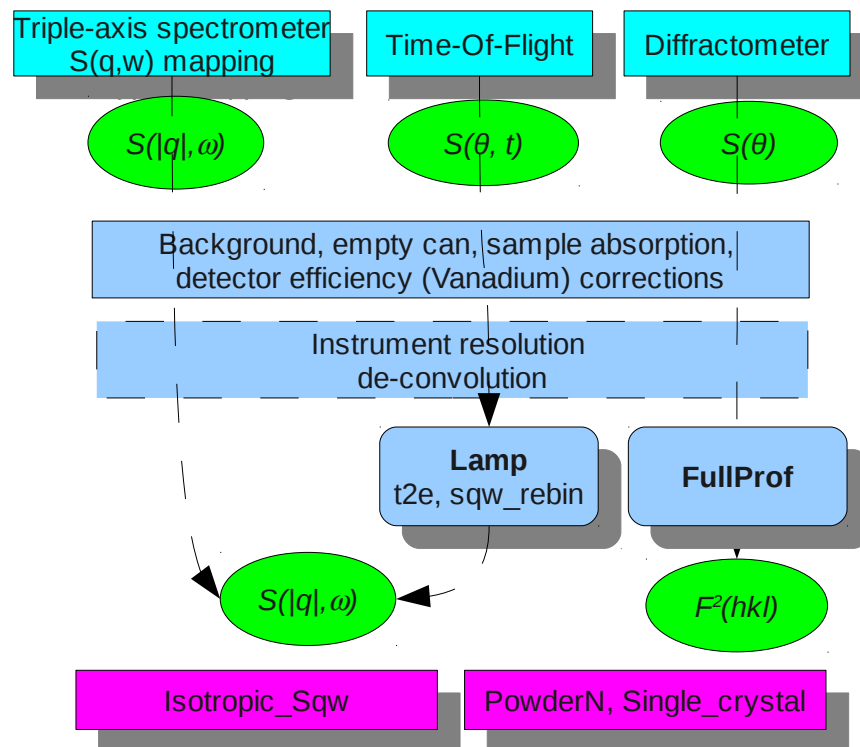


Figure 25. From experimental data to dynamical and structural information

Many samples have been measured with neutron scattering spectroscopy instruments since 50 years. Some of these results have been published, and in some cases the original data files are still available. In this case, a thorough data

analysis will provide the dynamical structure factor, as shown in *Figure 25*. This requires to subtract the empty can contribution, correct for detector efficiency and cell absorption. The time-of-flight data should be converted to momentum-energy space using e.g. the *LAMP* software [21]. The total dynamic structure factor is then obtained. In some cases, the use of polarized neutron beam experiments provides a way to separate the coherent contribution from the incoherent one, at the cost of measurement statistics. Triple-axis instruments may provide a direct measurement of a portion of the dynamical structure factor, by mapping the momentum-energy space. Instrument resolution corrections must then be accounted for, on top of the usual data corrections. The final data should then be formatted accordingly to be read by e.g. the *Isotropic_Sqw* component. Alternatively, powder and crystal structures may be refined using e.g. *Fullprof* [17] and formatted for use by the *PowderN* and *Single_crystal* McStas components.

8.2.2 From molecular dynamics

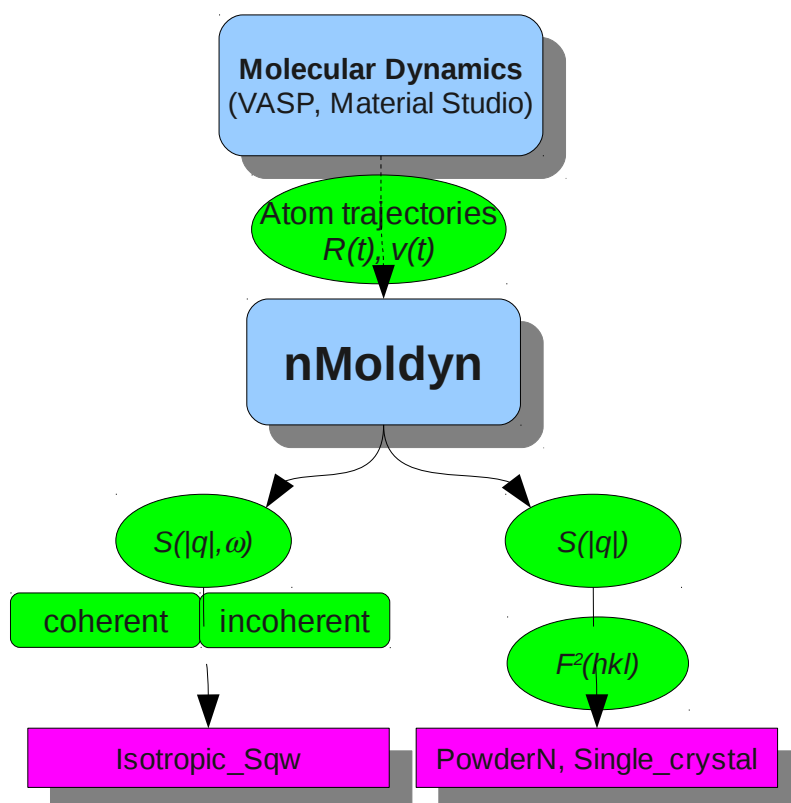


Figure 26. From molecular dynamics to dynamical and structural information

With ever increasing computer power, it has become quite usual to simulate complex materials using atomistic simulation codes, such as *Material Studio* [24], *VASP* [25], *AbInit* [26] or *Quantum Espresso* [27]. However, in order to obtain a full description of the material, including coherent and incoherent processes with elastic and inelastic contributions, a full atom trajectory must be simulated. Lattice dynamics codes such as *PHONON* [28] only provide the inelastic part. *Ab initio* (based on DFT) codes provide the most accurate results, but classical molecular dynamics (using parametrized interaction potentials) are required for larger systems.

The space-time atom trajectories should then be Fourier-transformed into a momentum-energy space to evaluate the dynamic structure factor. These operations may be performed using e.g. the *nMoldyn* package [22]. This procedure was used to produce the liquid rubidium [19] and germanium [23] dynamic structure factors included in the *Mcstas* data base.

9. CONCLUSION AND OUTLOOK

We have demonstrated a number of instrument simulation models, which may be assembled in order to exhibit non-trivial results for neutron scattering experiments. All of these models, even though rather concise, still produce data which compare with actual measurements. In particular, a number of measurement imperfections are reproduced, and help in understanding the required corrections and care that should be given during both the measurements, and the

further data analysis.

In the near future, such models will undoubtedly closely couple with the data analysis procedures to correct most of the instrumental effects and help to identify unknown features in measurement data. It may also become possible to evaluate the feasibility of submitted experiment proposals based on simple models.

References

- [1] N. Metropolis and S. Ulam. “*The Monte Carlo method*”. *Journal of the American Statistical Association* **44** (1949) 335-341.
- [2] F. James. *Rep. Prog. Phys.*, **43** (1980) 1145.
- [3] Grimmett, G. R., and Stirzaker and D. R. “*Probability and Random Processes*”, 2nd Edition. Clarendon Press, Oxford, 1992.
- [4] Marsaglia, George, “*Random number generators*”, *Journal of Modern Applied Statistical Methods*, **2** (2003) 2.
- [5] Matsumoto, M.; Nishimura, T. “*Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator*”, *ACM Transactions on Modelling and Computer Simulation* **8** (1998) 3–30.
- [6] K. Lefmann and K. Nielsen, *Neutron News* **10**, 20, (1999).
- [7] P. Willendrup, E. Farhi and K. Lefmann, *Physica B*, **350** (2004) 735.
- [8] K. Lefmann et al, “*Virtual experiments: the ultimate aim of neutron ray-tracing simulations*”, *Journal of Neutron Research*, **16** (2008), 97–111 .
- [9] McStas web-site <<http://www.mcstas.org>>
- [10] NeXus format web-site <<http://www.nexusformat.org>>
- [11] J. Peters, *Nucl. Instr. Meth. A* **540** (2005) 419.
- [12] M. Marseguerra and G. Pauli, *Nucl. Instr. Meth.* **4** (1959) 140
- [13] C. D. Clark, E. W. J. Mitchell, D. W. Palmer and I. H. Wilson “*The design of a velocity selector for long wavelength neutrons*”, *J. Sci. Instrum.* **43** (1966) 1
- [14] G.L. Squires, “*Introduction to the theory of thermal neutron scattering*”, Dover edition (1996) and Cambridge University Press (1978).
- [15] 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.
- [16] K. Yvon, W. Jeitschko and E. Parthé, “*LAZY PULVERIX, a computer program, for calculating X-ray and neutron diffraction powder patterns*”, *J. Appl. Cryst.* **10** (1977) 73 ; see also <<http://icsd.ill.fr>>.
- [17] J. Rodriguez-Carvajal, *Physica B* **192** (1993) 55.
- [18] V.F. Sears. *Adv. Phys.*, **24**, (1975) 1.
- [19] 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.
- [20] Theo Siegrist, *J. Appl. Cryst.* **30** (1997) 418.
- [21] D. Richard, M. Ferrand, G. J. Kearley, <<http://www.ill.fr/computing>>
- [22] T. Rog, K. Murzyn, K. Hinsén, G.R. Kneller, *J. Comput. Chem.* **24** (2003) 657.
- [23] V. Hugouvieux, E. Farhi, MR Johnson, et al., *Phys. Rev. B* **75** (2007) 104208.
- [24] *Material Studio* from Accelrys <<http://accelrys.com/products/materials-studio/>> San Diego, CA 92121 USA.
- [25] G. Kresse and J. Hafner. “*Ab initio molecular dynamics for liquid metals*”. *Phys. Rev. B*, **47** (1993) 558.
- [26] X. Gonze, B. Amador, P.-M. Anglade, J.-M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Cote, T. Deutsch, L. Genovese, Ph. Ghosez, M. Giantomassi, S. Goedecker, D.R. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M.J.T. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G.-M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M.J. Verstraete, G. Zerah, J.W. Zwanziger, *Computer Phys. Commun.* **180** (2009) 2582.
- [27] P. Giannozzi et al. *J. Phys. Condens. Matter* **21** (2009) 395502.
- [28] K. Parlinsky <<http://wolf.ifj.edu.pl/phonon/>>