

**Neutron Instrument Simulation Package
(NISP)**

Users' Manual *and* Tutorial

07 August 2006

(Revised 10 November 2009)

P. A. Seeger

239 Loma del Escolar, Los Alamos, NM 87544, U.S.A.

e-mail: PASEeger@losalamos.com

L. L. Daemen

Los Alamos National Laboratory, Los Alamos, NM 87545, U.S.A.

e-mail: lld@lanl.gov

Table of Contents

| | |
|---|-----------|
| Introduction. | 1 |
| I. How to Create an Instrument | 2 |
| I.A. Starting NISP_Win | 2 |
| I.B. Creating a Blank Instrument | 3 |
| I.C. Using a Previous Instrument as a Template | 3 |
| I.D. Offline Editing of an Instrument | 3 |
| II. How to Use “Help” | 4 |
| III. How to Define a Source | 4 |
| III.A. Extrinsic Parameters | 4 |
| III.B. Source Size and Phase Space | 5 |
| III.C. Pulsed or Steady-State Source | 6 |
| III.D. Choosing the Wavelength Distribution | 6 |
| III.D.1. Using a Table (.tbl) file | 6 |
| III.D.2. Using a Monitor (.mon) file | 7 |
| III.D.3. Sum of Maxwellian Distributions | 8 |
| III.D.4. Fixed Sources: Square and Triangular distributions | 9 |
| III.E. Source Polarization | 9 |
| III.F. OK / Cancel | 9 |
| IV. How to Define a Sample | 9 |
| IV.A. Small-Angle Scattering | 10 |
| IV.B. Reflectometry Sample | 10 |
| IV.C. Powder Diffraction | 10 |
| IV.D. Single Crystal Sample | 11 |
| IV.E. Inelastic Isotropic | 11 |
| IV.F. Deep Inelastic, Impulse Model | 11 |
| V. How to Define a Detector | 12 |
| V.A. Efficiency and Time-of-Flight Parameters | 12 |
| V.A.1. Logarithmic time scale | 12 |
| V.A.2. Linear time scale | 13 |
| V.A.3. No time scale | 13 |
| V.B. Plane Detectors | 13 |
| V.C. Cylindrical or Conical Detectors | 13 |
| V.D. Spherical Detectors | 14 |
| VI. Other Element Types in NISP_Win | 14 |
| VI.A. Materials – How to Cheat | 15 |
| VI.B. “Passive” Elements | 15 |
| VI.C. Choppers and Gravity Focuser | 16 |
| VI.D. Guides/Bender | 17 |
| VI.E. Mosaic Crystals | 17 |
| VI.F. Soller and Radial Collimators | 17 |
| VI.G. Using Special Surfaces to modify region geometry | 18 |
| VII. How to Fill Space Between Elements | 18 |
| VII.A. Sequential Elements | 18 |
| VII.B. Use of a Scattering Chamber | 19 |
| VII.B.1. Chamber (Cylinder) | 19 |
| VII.B.2. Chamber (Long) | 20 |
| VII.B.3. Chamber (Sphere) | 20 |

| | |
|---|-----------|
| VII.B.4. Identify Elements in the Chamber - Nesting | 20 |
| VIII. How to Visualize an Instrument | 20 |
| IX. How to Create Magnetic Fields | 20 |
| IX.A. Fields Specified with Fixed B and Interpolation | 21 |
| IX.B. Fields Generated by a Number of Current Loops, or Wire Segments | 22 |
| IX.C. One or Two Rectangular Loops | 22 |
| IX.D. Field of a Solenoid | 22 |
| X. How to Write Geometry Files | 22 |
| X.A. Multiple Files with Variable Parameters | 23 |
| XI. How to Execute MC_Run | 23 |
| XI.A. Initialization | 23 |
| XI.B. TRACE Mode | 25 |
| XI.C. Intermediate Output | 26 |
| XI.D. Visualizing and Analyzing Output | 26 |
| XI.D.1. Statistics of the Monte Carlo Simulation | 26 |
| XI.D.2. Simulated Data | 27 |
| XII. Tutorial Example: HIPPO | 28 |
| XII.A. Use NISP_Win to Define the Beam Elements | 28 |
| XII.B. “Connect” and Save Files | 31 |
| XII.C. Run the Simulation with MC_Run | 33 |
| XII.D. View Raw Data with See_MC_Data – Fix Problems | 33 |
| XII.E. Reduce Raw Detector Data to Histograms vs. d, DLfromZt | 33 |
| XIII. In Case of Difficulty | 36 |
| Appendix. Installation of NISP in a Local Computer | 37 |
| A. Download the Files | 37 |
| B. Extract the Programs from the .zip File | 37 |
| C. Source Files | 38 |
| D. Compiling with Visual Fortran | 38 |
| E. Analysis Programs | 39 |
| F. Data Files for the HIPPO Tutorial | 39 |
| G. Executing MC_Run in Batch Mode | 40 |

Introduction.

The Neutron Instrument Simulation Package (NISP) is a family of Monte Carlo codes to assist in the design and analysis of neutron beam lines and instruments. In particular, it should be used for comparison of design options; as is typical of Monte Carlo codes, precision is enhanced by running many histories, but accuracy is always limited by the algorithms used. It is our intention that the code be completely versatile and that it will continue to grow as users contribute new and improved code modules. Other uses of NISP include analysis of experimental resolution, optimizing parameters of an experiment, and teaching neutron scattering. Monte Carlo is not a substitute for thinking, but is a very appropriate tool for verifying design ideas and exploring novel concepts.

NISP has three parts: the user interface NISP_Win, the Monte Carlo engine MC_Run, and the Fortran subroutine library MCLIB (see Fig. 1). This Users' Manual describes how to use NISP_Win to generate a geometry file, and how to run a simulation with MC_Run. Instructions for the installation of NISP are given in the Appendix. Additional documentation of the element types in NISP is embedded in the code (see Sec. II). The MCLIB reference document ([Documents: MCLIB_2009.pdf](#)) describes the structures and philosophy of MCLIB and MC_Run, lists the modules in the library, and discusses how to implement a new *region* type. Other documents at the same location describe some of the more advanced algorithms used; the best source of that information is to be found in the Fortran source code itself, which is well commented and includes references where appropriate. Details on how to install a new *element* type in NISP_Win can be found in [Documents: SPIE_5536\(2004\)109.pdf](#).

[**NOTE:** throughout this Manual the link “[Documents:](#)” refers to <http://PASeeger.com/documents/>, and the link “[Downloads:](#)” refers to <http://PASeeger.com/downloads/>.]

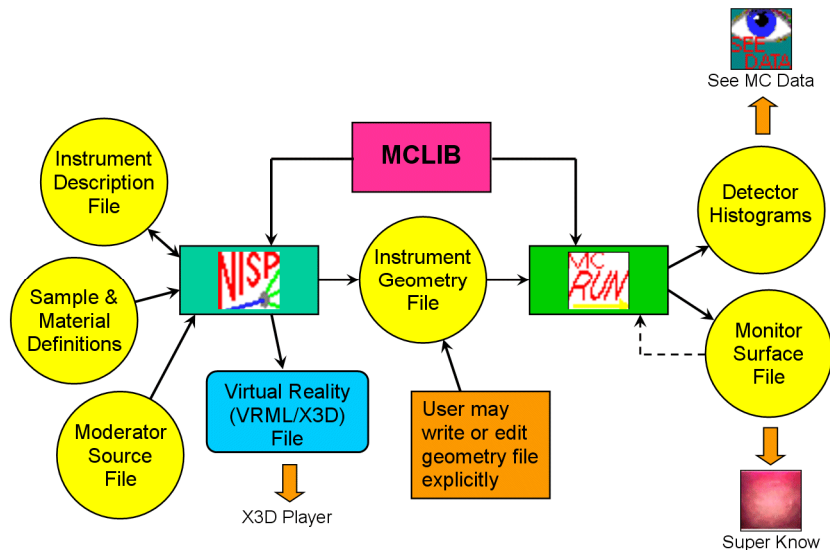


Figure 1. The components of NISP. This manual describes how to build an instrument using the NISP_Win interface, and then run the simulation using MC_Run.

MCLIB is a direct descendent of the subroutine library developed at the Rutherford Laboratory [M. W. Johnson & C. Stephanou, report RL-78-090 (Sept., 1978); M. W. Johnson, report RL-80-065 (Dec. 1980)]. The structure assumes all space is divided into “regions” by quadratic “surfaces,” and the interior of any region is described by a particular material or algorithm. This is the same philosophy as the well-known MCNP code for neutron and photon transport [J. Briesmeister (ed.), Los Alamos report LA-12212 (Nov. 1993)]. The fundamental difference between NISP and MCNP is that MCNP deals with microscopic *nuclear* interactions of neutrons in the regions, while NISP emphasizes *optical* properties (including surface interactions). NISP also includes gravity, and allows time-phased devices such as choppers. The concept of “region” is extended in NISP to form “elements” that may consist of one or more regions. The elements represent actual beam-line devices.

A unique feature of NISP among currently available codes is the transport of neutrons between *any* defined regions, rather than a strict linear sequence. For instance, scattered neutrons from any element may reach a detector, contributing to background. Neutrons may be split into transmitted and scattered components, or reflected and not reflected, or parallel and anti-parallel spins, and the resulting histories will follow completely different paths. Inclusion of magnetic fields and spin precession is also unique.

We actively request input to NISP. Please send all questions, comments, and suggestions by e-mail to lld@lanl.gov and to PASeeger@losalamos.com. In particular, we will be happy to help in the incorporation of your code as new region and element types.

I. How to Create an Instrument

Instruments to be simulated are built using the user interface, NISP_Win. Before executing NISP_Win you should make a detailed sketch of the instrument. You will need world coordinates of every element in the beam line, typically at the center of the entrance surface of the element. We frequently define the origin of the coordinate system to be at the sample, but it may also be at the moderator surface, or at a monochromator. Fig. 2 shows the coordinate system. The Y-axis is *always* vertical, Z is generally the nominal beamline direction, and X is measured to the neutron's right as it moves in the Z-direction. (Note that the system is left-handed.) You will be allowed to change the direction of the Z-axis, for instance at a monochromator or bender, in which case the Z-coordinate is cumulative. If a beam splits into two parts (*e.g.*, at a polarizer) then both components must be defined in the same coordinate system.

You also need the orientation of each element with respect to the world coordinate system, using the angles defined in Fig. 3. The axis of the element may have a Tilt in the horizontal plane, and/or a Slope out of the plane. In a few cases, elements that are not circularly symmetric about their axes may also need the Rotate angle. See [Documents: TiltSlopeRotate_2005.pdf](#) for details of the coordinate transform matrices.

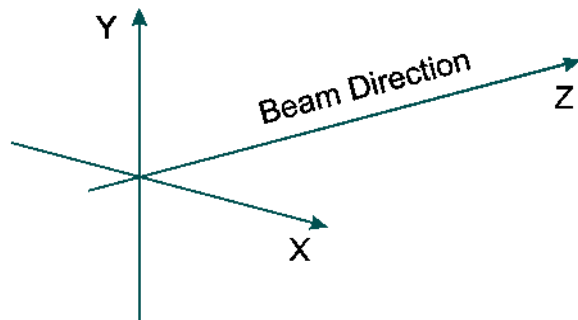


Figure 2. NISP coordinate system. Y is always vertical, and the Z-axis is the nominal beam direction. The coordinate system is left-handed.

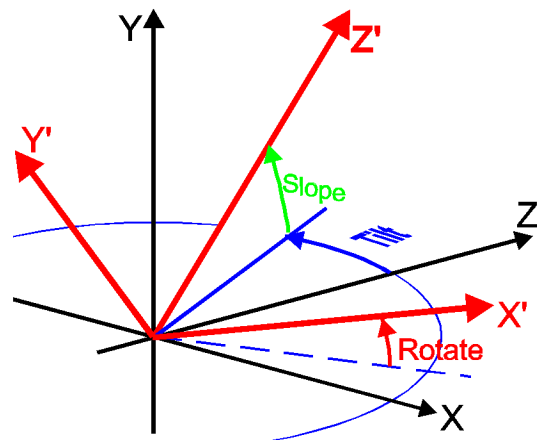


Figure 3. Orientation angles, Tilt, Slope, and Rotate. Z' is the symmetry axis of the element, and (X, Y, Z) are world coordinates.

I.A. Starting NISP_Win

The most convenient way to execute NISP_Win is from a desktop icon. You can set the “Properties” of the icon to “Start in:” the directory where you wish your instrument and geometry files to be stored (right-click the icon to set properties). The first screen that opens is a Help message giving the code version and date, the required data files, and a description of the five entries on the Menu bar:

- **File:** New, Open, Save, Save As, Geometry, Multiple Geom, Exit; create and save files; Sec I and X.

- **Elements:** Add Element, Edit Element, Delete Element, Vary Parameter, View Elements, List Variables; fundamental operations to build and edit instrument files, Sec. III through VI.
- **Connect:** Enable; turn on/off the connection-matrix window; see Sec. VII.
- **3D View;** write a Virtual Reality file for the instrument; see Sec. VIII.
- **Help:** General Help, Element Help, QuickWin Help, About; see Sec. II.

You can return to this message at any time by clicking Help > General Help. Clear by typing any key.

I.B. Creating a Blank Instrument

Click File > New. In the window that opens, navigate to the appropriate folder and enter the file name. The default extension is “.inst”. If you use an existing file name you will be asked to confirm, because the old file will be overwritten. You will then be asked to enter a descriptive title that will appear on the text output files. This name can be rather generic, as the name specific to a given case will be entered later in the Source element.

I.C. Using a Previous Instrument as a Template

Often, the best way to start is by editing an existing instrument. In particular, if you are studying various configurations of the same instrument you may only be changing some parameters or a small number of elements. Click File > Open and choose an existing file; either a .inst or a .geo file may be used. You may keep the old name or enter a new one. Unless you wish to overwrite the old .inst file, you should File > Save As with a new file name.

I.D. Offline Editing of an Instrument

You may use a text editor to modify a .inst file. This is useful for a global change of a value, or if you have added new elements (always appended to the end of the file) and want to arrange them in a more logical order. If you do anything that changes the number or the order of elements, you should delete the connection matrix (everything following the line with (.end)).

Glossary

Some terms used in NISP have specialized meanings.

- **neutron:** a structure with 13 parameters giving position (X,Y,Z), velocity (Vx,Vy,Vz), time (TOF), mass (M=1), charge (Q=0), statistical weight (WT), and polarization vector (Px,Py,Pz).
- **statistical weight:** number (or fraction) of particles represented by a neutron structure; used to track less probable events. A neutron may be attenuated in a material by having its weight reduced. When a neutron splits, for instance into two spin states, the sum of final weights will equal the initial weight (for normalization).
- **history:** the sequence of events for a neutron, including surface contacts, region entries, and interactions within regions, starting at the source *or* following a split. The number of histories reaching a detector bin determines the statistics in that bin, while sum of weights is intensity.
- **surface:** a general quadratic expression of the form $Ax^2+Bx+Cy^2+Dy+Ez^2+Fz+G+Pxy+Qyz+Rzx$; a neutron is “on” the surface if the expression evaluates to 0. Surfaces also have a roughness parameter, BETA.
- **sense:** a neutron for which the surface expression is >0 is on the + side (and <0 on the - side) of the surface. If the expression is 0, the sign of the 1st derivative of the expression is used, and if that is also 0, then the sign of the 2nd derivative (or 0) is used for sense.
- **region:** a volume of space bounded by surfaces. The interior of the region is defined by the sense of each bounding surface. Each region has a name.
- **material:** the contents of a region may be a *physical* material, or a more general *algorithm* in the MCLIB library. When a neutron passes through a region, the material determines what happens to it. When it reaches a surface, the material on the other side is tested to determine if a surface interaction occurs.
- **element:** an actual piece of a beam line, consisting of one or more regions and their materials, selected and defined by user input in NISP_Win.
- **connection matrix:** an array showing which elements are to be connected to each other, either sequentially or by being surrounded.
- **geometry file:** a structure with all of the surfaces, regions, and material parameters, in a format to be read by the Monte Carlo program MC_Run.

II. How to Use “Help”

The built-in Help can be accessed either when editing an element or by clicking Help > Element Help when NISP_Win is running, **or** by executing `\NISP\Help\NISP.hlp` at any time. In the latter cases you can select an element type from an alphabetical list, *or* you can create an index and *search* for all element references and help entries that contain any particular word. The help entry for each element type has a description of the element and a set of links to the descriptions of the element parameters.

When you are building an instrument and have selected an element type to add or to edit, a dialog box will come up for entering parameters. Next to every parameter is a small box with “?” (*cf.* Fig. 4). Clicking that box will bring up the appropriate help message. When you click the “?” next to the Element Name box on the top line you will get the description of the element.

We try to keep the descriptions clear and accurate, and frequently add additional information to existing entries, such as examples of usage. If you find any descriptions to be confusing, ambiguous, or apparently wrong, or any links that take you to the wrong parameter, we need your input! Please remember that the help messages were written and proofread by people who already knew what the parameters are. Only *you* can tell us whether they are actually helpful. Please e-mail any suggestions to PASeeger@losalamos.com.

III. How to Define a Source

Every instrument must have a source of neutrons. From the menu bar, click Elements > Add Element > Source (Plane) > OK. An instance of a Source (Plane) will be created for your instrument. Fig. 4 shows the screen for entering the parameters of the Source (Plane). There are four different ways to define the source: Table file, Monitor file, sum of Maxwellians, or Fixed wavelength distributions. Sources may also be either pulsed or steady-state, and may set the polarization vector. This is the most complex of all element template!

Since every model has one and only one source, the name entered for the source is used as the Instrument Name. Enter a descriptive name (up to 40 characters) in the Instrument Name box at the top of the screen. This name will be carried through all data files, so it should be unique to this instrument. When running many cases, try to have enough information in the instrument name to distinguish them.

III.A. Extrinsic Parameters

Element parameters are divided into “Extrinsic” and “Intrinsic.” The possible extrinsic parameters are (X, Y, Z, Tilt, Slope, Rotate), as defined above (Figs. 2, 3) in world coordinates. All other parameters of any element are intrinsic, defined only within the element. The element may be moved without changing intrinsic parameters. At the time a geometry file is written all surfaces are converted automatically to world coordinates.

When entering a value in a parameter box as in Fig. 4, note the *units* written to the left of the box. Generally the transverse dimensions (X and Y) will be mm, but the Z coordinate is m. The coordinates for a Source (Plane) are the center of the surface area being viewed. You need to choose where to place the origin of the coordinate system. One choice is at the moderator surface, in which case you will enter “0” for Z. However if the geometry of the secondary flight path(s) is complex, it will be easier to define the origin at the sample, in which case Z for the source will be negative.

For pulsed sources the reference plane of the source surface is also the reference for time distributions. That is, pulse shapes are defined as the times when neutrons cross the surface, and it is important to account for the Tilt angle of the source plane. Intensity is also weighted by the cosine of the off-normal angle. For steady-state sources the location and orientation of a reference surface are somewhat arbitrary.

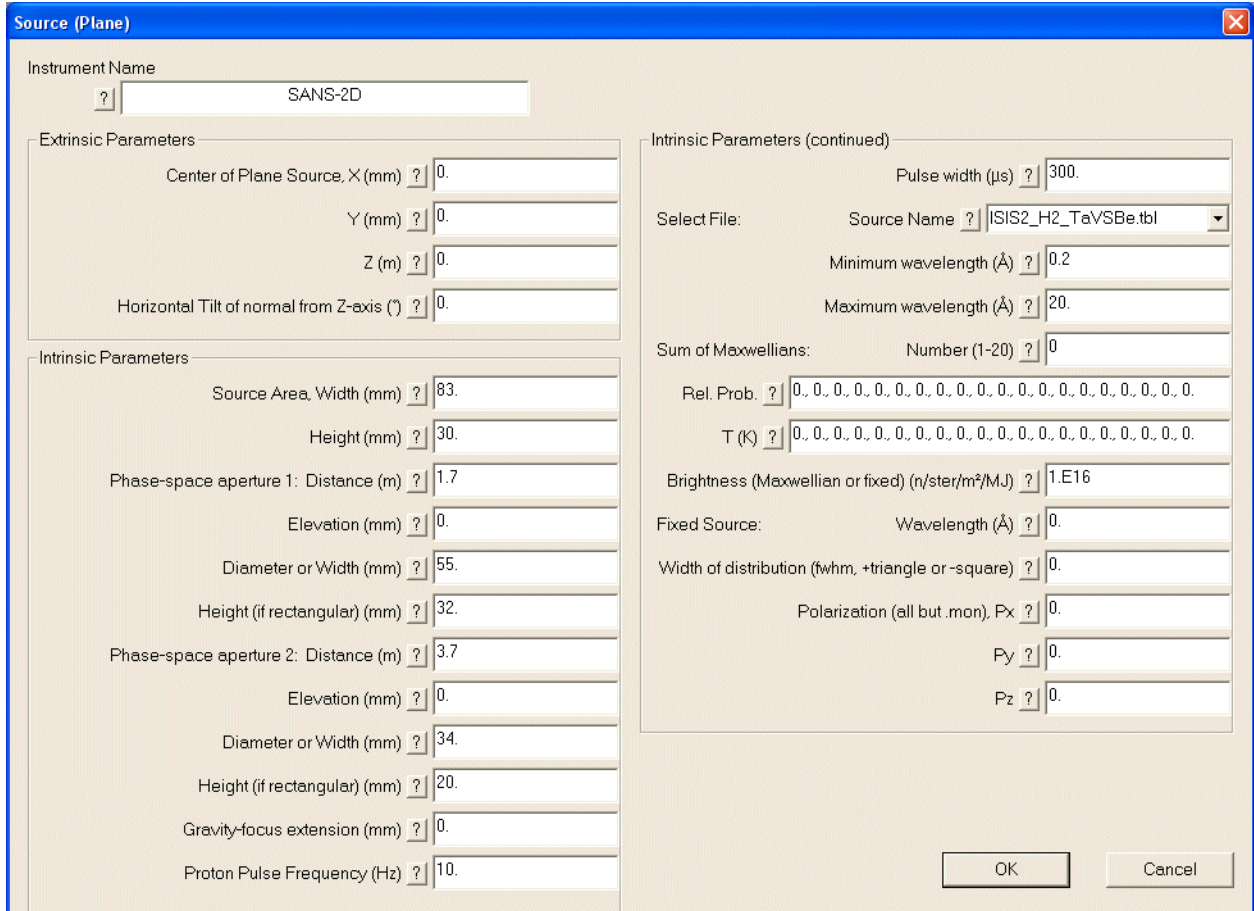


Figure 4. NISP_Win dialog box for entering parameters of element type Source (Plane). This is the most complex element in NISP, since it includes moderator geometry, phase space to be sampled, and alternate parameters for several source algorithms (described in the text). Note that every entry has a “?” to click for help. The example shown uses table interpolation for the H₂ moderator at ISIS Target 2.

III.B. Source Size and Phase Space

The source is a plane rectangular surface with full width and full height given in mm. The phase space to be sampled is defined by two virtual apertures, as shown in Fig. 5. The parameters of each aperture are its Z- and Y-positions measured with respect to the center of the moderator surface, its diameter or full width, and for a rectangular aperture its full height. (For a circular aperture, leave the ΔY term zero.) Either or both apertures may represent actual apertures in the instrument. There are two ways to define the phase space. Fig. 5a shows the umbra and penumbra of the “usual” 2-aperture case. If the instrument begins with a neutron guide, however, it may be

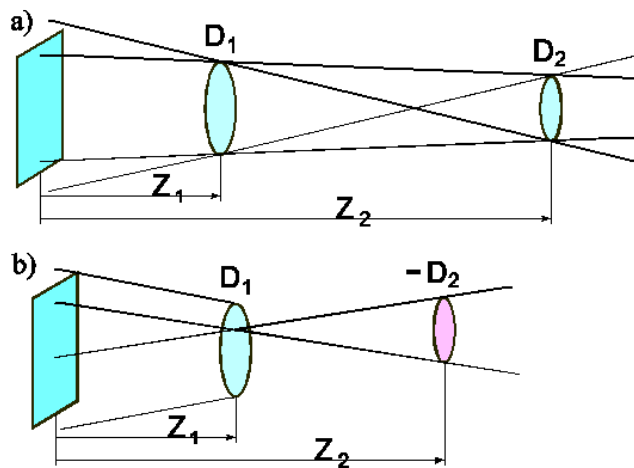


Figure 5. Phase-space definitions. a) usual 2-aperture case. b) when D₂ is negative, 1st aperture defines area and 2nd defines solid angle.

preferable to separate the definitions of the area and angular parts of the phase-space volume. As illustrated in Fig. 5b, this option is flagged by setting the sign of the diameter (width) of the second aperture negative. The first aperture is fully illuminated, but only neutrons within the solid-angle cone (or pyramid) defined by the second aperture as seen from the first are sampled. For example, suppose you have an $m=3$ Ni supermirror guide with effective critical angle $1.7 \text{ mrad}/\text{\AA}$, and you want to include all $4\text{-}\text{\AA}$ neutrons. The half-angle of the cone should be 6.8 mrad , so set $D_2 = -13.6 \text{ mm}$ and $Z_2 = Z_1 + 1 \text{ m}$.

In both cases the total phase space sampled is the product of the areas divided by the square of the separation of the apertures. If the second aperture has zero area, phase space will be mm^2 instead of $\text{mm}^2\text{-ster}$.

Note that the apertures may be raised above the centerline (positive values of Y_1 and Y_2) to correct for gravity to view the centers of the moderator and sample. If a gravity focuser is used to bring all wavelengths to the center of the detector, then the parameter “Gravity-focus extension” should be set to the stroke length of the gravity focuser; otherwise leave that parameter zero.

III.C. Pulsed or Steady-State Source

The number used as a flag for a pulsed source is the “Proton pulse frequency (Hz).” The name of the parameter refers to a spallation neutron source at a proton accelerator; for a steady-state (reactor) source set the frequency to 0. The “Proton pulse width (μs)” parameter also refers to a spallation source. These parameters affect only the time distribution of neutrons, and are only relevant for Table-file sources.

III.D. Choosing the Wavelength Distribution

There are four methods for defining the wavelength distribution, which use different blocks of the parameters in the second column of the dialog box of Fig. 4, as highlighted in the following sections.

III.D.1. Using a Table (.tbl) file

Parameters used are **Select File**, λ_{min} , and λ_{max} . The remaining parameters in Fig. 4 are ignored if a .tbl file is selected.

Pulsed neutron sources are generally under-moderated and have an epithermal component, making them difficult to define precisely with Maxwellians or other standard distribution functions. For these sources, NISP samples a tabulated distribution. To use a Table file for the neutron source, select the name from the pull-down menu. Use λ_{min} and λ_{max} to limit the range sampled from the table.

A large number of spectra computed from MCNP for various target-moderator systems at spallation sources are available in files, and the most important are listed in Table 1. The latest files are sampled by reverse interpolation in a cubic spline fitted to the cumulative distribution weighted by λ . The procedure to generate the tables is given in [Documents: SourceFiles_2005.pdf](#). A typical energy spectrum is shown in Fig. 6, for the SNS water moderator facing flight path 17. The blue histogram is the input “data” from MCNP, the red dots are the spline nodes, and the red line is the inverse interpolation.

The spectrum files for pulsed sources also include parameters for the wavelength-dependent pulse shapes. Older files use various analytic forms fitted to the MCNP data. New files use the same kind of spline interpolation as for energy. Separate splines are found for the epithermal and thermalized components and a linear switch function is used for intermediate wavelengths. Fig. 7 shows such an interpolation, for the same source as Fig. 6. The black points are from MCNP, and the brown curve is the linear combination of the epithermal (red) and thermal (blue) shapes.

There are files for three steady-state sources. D11 measured with the Brünhilde velocity selector on the ILL cold source, and two files representing the spectra from the SINQ steady-state spallation source. These files have no pulse-shape information, so the Proton Pulse Frequency should be set to 0.

Table 1. Selected Table files for NISP

| Facility | .tbl File Name | Title | MCNP ID |
|--------------|----------------------|--|-------------------|
| Lujan | LujanII_01_HiResH2O | Lujan Mark-II, f.p. 16,1,2, HiRes H2O | GM_030709_000000 |
| | LujanII_04_H2O | Lujan Mark-II, f.p. 3-5, H2O | GM_030709_000000 |
| | LujanII_07_H2O | Lujan Mark-II, f.p. 6-8, H2O | GM_030709_000000 |
| | LujanII_10_dcplH2 | Lujan Mark-II, f.p. 9-11, Liq. H2 | GM_030709_000000 |
| | LujanII_12_cpdlH2 | Lujan Mark-II, f.p. 12-13, Coupled H2 | GM_030709_000000 |
| | LujanII_14_upH2O | Lujan Mark-II, f.p. 14-15, Upper H2O | GM_030709_000000 |
| | | | |
| IPNS | IPNS_C_groovedCH4 | IPNS "C" (solid CH ₄ , 30K, grooved) | EBI_000801_093603 |
| | IPNS_F_liqCH4 | IPNS "F" (liquid CH ₄ , 100K, poisoned) | EBI_000801_093603 |
| | IPNS_H_solCH4 | IPNS "H" (solid CH ₄ , 30K, poisoned) | EBI_000801_093603 |
| | | | |
| ISIS | ISIS1W_N5_liqH2 | ISIS Target1, W, Liquid H2 | SA_030609_152400 |
| | ISIS1W_S2&N8_H2O | ISIS Target1, W, Upstream H2O | SA_030609_152400 |
| | ISIS1W_S8&N2_CH4 | ISIS Target1, W, CH4 | SA_030609_152400 |
| | ISIS2_Broad_TaVSBBe | Broad, ISIS Tgt2, Ta vessel/solid Be | SA_040615_094100 |
| | ISIS2_Groove_TaVSBBe | Groove, ISIS Tgt2, Ta vessel/solid Be | SA_040615_094100 |
| | ISIS2_H2_TaVSBBe | Hydrogen, ISIS Tgt2, Ta vessel/solid Be | SA_040615_094100 |
| | ISIS2_Narrow_TaVSBBe | Narrow, ISIS Tgt2, Ta vessel/solid Be | SA_040615_094100 |
| | | | |
| SNS | SNS_sct_bu08_15H2O | SNS-sct521_bu_08, 15-mm H2O | EBI_020327_175548 |
| | SNS_sct_bu17_25H2O | SNS-sct521_bu_17, 25-mm H2O | EBI_020327_175548 |
| | SNS_sct_td05_cH2 | SNS-sct21a_td_05, Coupled H2 | EBI_020327_170520 |
| | SNS_sct_td14_cH2 | SNS-sct41a_bd_14, Coupled H2 | EBI_020327_175525 |
| | SNS_sct_tu02_dH2 | SNS-sct091_tu_02, decoupled H2 | EBI_020327_164257 |
| | SNS_sct_tu11_dH2 | SNS-sct091_tu_11, decoupled H2 | EBI_020327_164257 |
| | SNSL_fwdS | SNS-LWTS Front Wing, decoupled solid CH4 | EBI_000829_111030 |
| | SNSL_ssdL | SNS-LWTS Starboard Slab, liquid CH4 | EBI_000829_111030 |
| | SNSLpscS_new | SNS-LWTS Port Slab, coupled solid CH4 | EBI_000829_111030 |
| | | | |
| ILL | D11 | D11 n-A ² /ster/m ² /s | measured |
| | | | |
| SINQ | SINQ_cold_D2 | SINQ, Cold D2 | estimated |
| | SINQ_ambient_D2O | SINQ, Ambient D2O | estimated |

III.D.2. Using a Monitor (.mon) file

When the **Select File** specifies a monitor file, *all* other source parameters are ignored.

A monitor file may be written during execution of MC_Run. The file contains the histories of all neutrons crossing some surface of the problem. Such a file can be used as a source in subsequent simulations by entering its name in the "Enter Source Name" box. You should enter the name including the path to the file in your local computer. The file extension .mon is required. Note that monitor files are direct-access binary, and generally are *not* transportable between platforms.

A particular use of monitor files is in optimization of downstream elements. On the first run, monitor the neutrons that make it through the initial (constant) part of the instrument, and use those same neutrons as the source for other configurations.

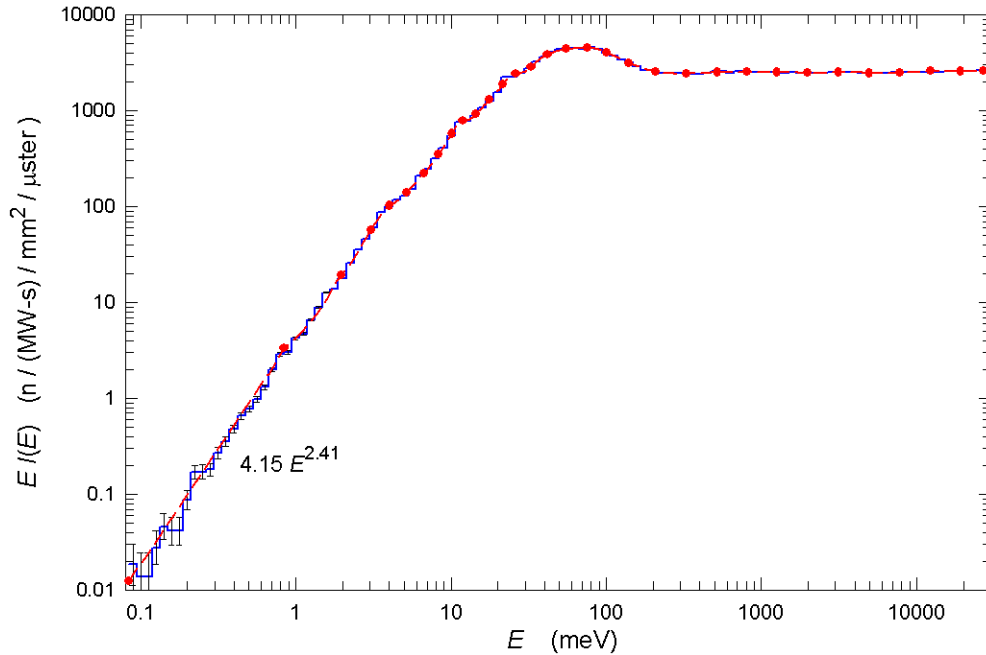


Figure 6. Energy spectrum from the SNS bottom-upstream water moderator (poisoned at 25-mm depth). The red dots are nodes of a spline curve fitted to the cumulative distribution of λ times the blue histogram of the MCNP simulation. The distribution is sampled by spline interpolation in the cumulative distribution.

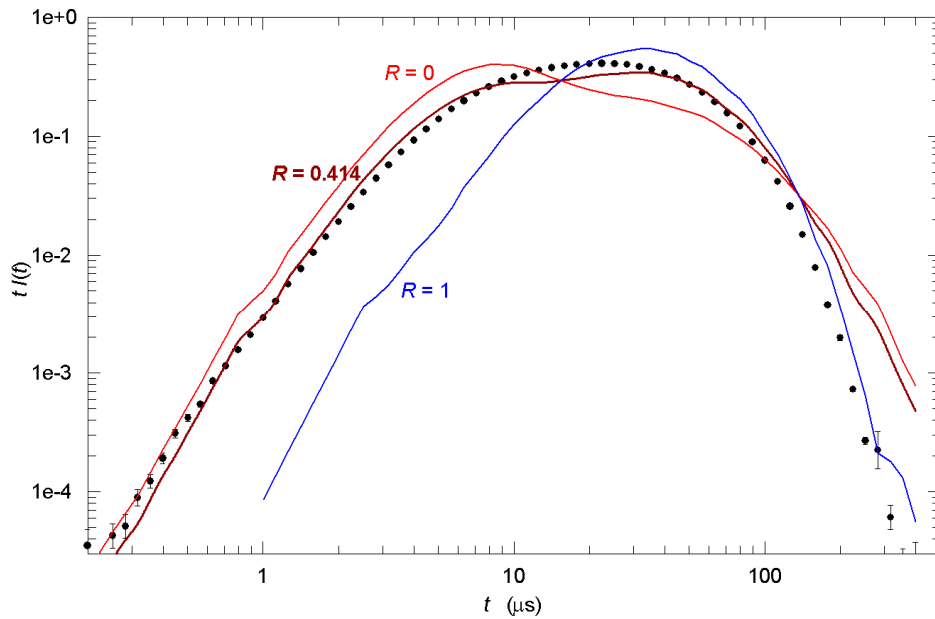


Figure 7. Time distribution at an intermediate wavelength ($\lambda = 1.2 \text{ \AA}$) for the SNS 25-mm water moderator. The red and blue curves are respectively the epithermal and thermalized components, and the brown curve is a linear combination given by a linear switch function. Black dots are the data from an MCNP simulation.

III.D.3. Sum of Maxwellian Distributions

The parameters used are **Source Name**, **Number** of Maxwellians, **Relative Probability** and **Temperature** of each, and the combined integrated **Brightness**.

A steady-state source may be adequately represented by a weighted sum of Maxwellian distributions, typically two or three. If neither a .tbl nor a .mon file is used for the Source Name, then NISP_Win will

check the “Sum of Maxwellians: Number (1-20)” entry. (All array inputs in NISP_Win have a maximum of 20 values.) For each included Maxwellian, enter a relative probability in the list “Rel. Prob.” and the temperature parameter in the list “T(K).” The relative probabilities do not have to be normalized to 1, so numbers proportional to the intensities may be used. The actual integrated intensities of the Maxwellians must be added up and entered as “Brightness” with units $n/(\text{ster}\cdot\text{m}^2\cdot\text{MJ})$, normalized to the energy used to produce the neutrons (MW-s or MJ) to facilitate comparisons among spallation sources and reactors. For instance, at SINQ with proton energy 570 MeV, 1 mA-s is equal to 0.57 MJ, so intensities quoted per mA-s would be divided by (0.57 MJ/mA-s).

It is also possible to make a `.tbl` file that has the Maxwellian parameters instead of a look-up table, by examining the parameter definitions for a type 91 region in the `mc_elmnt.inc` file.

III.D.4. Fixed Sources: Square and Triangular distributions

The parameters used are **Source Name**, **Brightness**, **Wavelength**, and distribution relative **Width**. For a fixed wavelength or simple distribution, set λ_0 to the center wavelength. A positive value of $\Delta\lambda/\lambda$ is the full width at half-maximum of a triangular distribution, such as would be produced by a velocity selector. For a δ -function, set $\Delta\lambda/\lambda$ to 0. A negative value gives a square distribution. When you use one of these distributions, you should enter a descriptive name in the Source Name box. You can generate a square *time* distribution by giving a value for Proton pulse width.

III.E. Source Polarization

Any source (except a Monitor file) may also be polarized in any direction by entering the three components of a polarization vector. **P** gives the direction and $|\mathbf{P}|$ the degree of polarization. If $|\mathbf{P}|$ is greater than 1, then **P** will be renormalized so that $|\mathbf{P}| = 1$.

III.F. OK / Cancel

When all parameters are satisfactory, click the “OK” button at the bottom of the screen. To exit the dialog box without creating (or editing) an element, click “Cancel”.

IV. How to Define a Sample

A complete instrument simulation requires one sample. You may run without a sample to simulate a beam profile or to create a monitor file. (You may also run with more than one sample for simultaneous simulation of two kinds of scattering, but care must be taken!) There are many sample types available, and additional algorithms are solicited. To add a sample to the open instrument file, click Elements > Add Element. Choose the sample type from the list in Table 2, and click OK. Begin by giving a descriptive Element Name, and finish (after all parameters have been entered) by clicking the “OK” button at the bottom of the screen. Some of the type-dependent parameters are described in the following sections. Sample regions cause neutrons to be split into “transmitted” and “scattered” components, and multiple scattering is generally applied (for a complete discussion of multiple scattering, see [Documents: NIMA_510\(2003\)290.pdf](#).)

Many sample types allow the outgoing solid angle to be limited, usually by specifying a range of azimuthal angles, to speed execution. You should be aware that using this feature turns *off* multiple scattering. Another feature being added is Polarizable samples. If a non-zero value is entered for “Polarization Saturation Field (T)” and if the sample is in a magnetic field region, it will be polarized in

Table 2. Available Sample types

| |
|---------------------------------------|
| Small-Angle Sample (Circle) |
| Small-Angle Sample (Rectangle) |
| Reflectometry Sample |
| Diffraction Sample (Flat) |
| Diffraction Sample (Cylinder) |
| Single Crystal Sample |
| Inelastic Isotropic Sample (Flat) |
| Inelastic Isotropic Sample (Cylinder) |
| Deep-Inelastic Sample (Flat) |
| Deep-Inelastic (Cylinder) |

the local **B** direction (or opposite if the parameter is negative) with efficiency $[1 - \exp(-|B|/|B_s|)]$. A small value of the Saturation parameter makes the sample “easy” to polarize.

IV.A. Small-Angle Scattering

Although originally intended for simulation of scattering at low momentum transfer (low Q), these types are useful for fixed-Q elastic scattering at any angle. The extrinsic parameters define the center of the entrance surface of the sample. In a true small-angle instrument, the sample would be flat. Tilt and Slope may be specified for wide-angle configurations. Thickness is measured perpendicular to the face of the sample, and “Sample Transmission at 10 Å” is also the transmission through the perpendicular distance assuming the *only* process removing neutrons from the beam is small-angle scattering. That is, the transmission parameter is used to estimate the fraction of beam *scattered*, which is proportional to λ^2 . We have chosen 10 Å as the reference wavelength because small-angle experiments tend to use long-wavelength neutrons.

Two scattering algorithms are available: constant Q, or uniform spherical particles with rigid surfaces. If no value is given for the “Hard Sphere Radius,” then every neutron will scatter at exactly the given value of momentum transfer Q, at an angle θ given by $\sin(\theta/2) = Q\lambda/4\pi$. If a value is entered for R, then the Rayleigh scattering law is sampled to find Q and thus θ .

IV.B. Reflectometry Sample

The geometry of a grazing-angle reflectometry sample is a horizontal plane that may have a Slope; *i.e.*, a plane parallel to the X-axis. (Vertical surfaces reflecting to the right or left are not implemented in NISP_Win.) The extrinsic X, Y, and Z define the center of the reflecting surface, and the intrinsic Width and Height parameters define its extent.

Reflectivity as a function of neutron wavelength is computed by optical interference from multiple layers with differing indices of refraction. This is a “box” model with up to a maximum of 20 discrete uniform layers. First describe the substrate and state how many layers are deposited on it. Then enter the layer parameters in lists, counting outward from the substrate. Note that the thickness of the substrate is given in mm, while the deposited layer thicknesses are in Å. Enter the real and imaginary parts of the scattering-length densities in units of $10^{10}/\text{cm}^2$ (which are the same as units of $10^{-6}/\text{Å}^2$). The Roughness parameter (rms Å) refers to the outer surface of the layer. It is used to represent fluctuations in layer thickness, or may be used to simulate diffusion.

IV.C. Powder Diffraction

The extrinsic parameters X, Y, and Z are the *center* of the sample (rather than on a surface). The Flat sample may also have Tilt and Slope parameters (Fig. 3). The Cylinder is always vertical, and it is possible to define a cylindrical *shell* by giving a value for the Inner Diameter that is positive and less than the outer diameter.

The sample material is usually selected from a list of files; use the pull-down menu in the “Powder Sample” box to see the currently available powders in the `\powders` folder. You may add files to the folder. The files contain cross section terms, and a number of Bragg edges and cumulative scattering cross sections computed with the Lazy-Pulverix general powder-pattern code [version of 21 Jul 1987, A. D. Murray & A. N. Fitch; see also K. Yvon, W. Jeitschko, & E. Parthe, *J. Appl. Cryst.* **10** (1977) 73–74]. Pulverix has been modified to write a file in our standard format ([Documents: DataFormat_1999.doc](#)), but parameters 2–6 in the second record do not have the usual meanings. The parameters are:

- 1) number of Bragg edges (number of entries in the POINTS and DATA blocks)
- 2) limiting (short-wavelength) macroscopic total cross section (1/cm)
- 3) macroscopic incoherent scattering cross section (1/cm)

- 4) macroscopic coherent scattering cross section at 1 Å, proportional to $1/v$ ($1/\text{cm}\cdot\text{Å}$)
- 5) macroscopic absorption cross section at 1 Å, proportional to $1/v$ ($1/\text{cm}\cdot\text{Å}$)
- 6) unit cell volume ($\text{b}\cdot\text{cm}$ or Å^3); this parameter is optional and is not copied to the geometry file.

The “POINTS” block in the file is the list of Bragg d-spacings (Å) in decreasing order, and the “DATA” block has the corresponding cumulative macroscopic cross sections at 1 Å ($1/\text{cm}\cdot\text{Å}^2$).

The file called `delta.pdr` is a special case useful for simulating resolution. It has nine equally weighted δ -functions, at $d = 0.15, 0.25, 0.5, 1, 2, 3, 4, 6,$ and 8 Å.

IV.D. Single Crystal Sample

The algorithm for single-crystal diffraction has been imported from McStas with the assistance of Kristian Nielsen [McStas rev. 1.7, 24 Apr 2003, <http://neutron.risoe.dk/documentation/manual/mcstas-1.4-manual.ps.gz>]. The Laue data files (`\Crystals*.lau`) have been edited so that NISP_Win can read the unit-cell parameters found on the second line using the namelist block “`cell`”. For instance, the parameters for the monoclinic crystal Adrenaline are

```
$cell a=7.873, b=6.790, c=8.638, aa=90, cc=90, bb=98.02/
```

where the lengths are in Å and angles in °. Any added files must be edited to have this same format.

The extrinsic location is the *center* of the sample. The unit-cell orientation prior to application of the Tilt-Slope-Rotate angles (Fig. 3) is that *a* coincides with +X and *b* is in the X-Y plane.

IV.E. Inelastic Isotropic

The extrinsic parameters X, Y, and Z are the *center* of the sample (rather than on a surface). The Flat sample may also have Tilt and Slope parameters (Fig. 3). The Cylinder is always vertical, and it is possible to define a cylindrical *shell* by giving a value for the Inner Diameter that is positive and less than the outer diameter.

The parameter “Transmission at 1 Å” is used to determine the wavelength-dependent scattering probability, and multiple scattering. It is the intensity of the attenuated beam after passing through a length of material equal to the perpendicular thickness of the flat sample, or the difference of outer and inner diameters for the cylinder (*i.e.*, the thickness of material on a diameter). A value near 1 will minimize effects of multiple scattering. A value *equal* to 1 is a special case that will result in 50% scattering at all wavelengths and no multiple scattering.

Up to 20 discrete energy levels may be entered. Entering 0 for the number of levels will give elastic scattering only; otherwise, the level parameters are entered as lists of E-gain (meV), relative probability, and statistical width (meV, positive for Gaussian and negative for Lorentzian). Note that E-gain will most often be *negative* because the neutron loses energy.

Normally, scattering is isotropic into 4π solid angle, but the solid angle may be limited by specifying the minimum and maximum scattering angle projections on each of the 3 principle axes. For instance, a detector bank on the left side of the beam direction could have minimum x-angle = 90° and maximum x-angle 180° . If the detector is 10° tall centered on the horizontal plane, then minimum y-angle = 85° and maximum y-angle = 95° . If it subtends scattering angles (2θ) from 15° to 165° , use those values for minimum and maximum z-angles. Applying these 3 limits, the sampled solid angle for this example is 3.5% of the sphere, so data collection is 28.5 times faster. Note that if this option is used, then multiple scattering is disabled. Also, because of the overlapping nature of the limits, you have to be careful not to eliminate *all* of the allowed scattering space!

IV.F. Deep Inelastic, Impulse Model

Flat or cylindrical geometry as above. See [Documents: Impulse_2008.pdf](#) for the algorithm, and use Help for the parameter definitions.

V. How to Define a Detector

The primary output of a simulation is a file with histograms of detector counts. There is no limit to the number of detectors in an instrument file, **but** the current version of MC_Run is limited to 20 banks. Each bank can be encoded in as many as 3 dimensions, but again MC_Run has a limit (2×10^6) on the total memory available for histograms. (The limits may be changed by recompiling MC_Run with new values of MAX2D and/or MAXDET in line 17.) Time will be recorded for pulsed source, or λ for steady-state source. Four detector geometries are available: plane circular, plane rectangular, conical (including cylindrical) surface, or spherical surface. To add a detector to an open instrument file, click Elements > Add Element, select the type, and click OK. Give the detector a distinctive Element Name to identify its output histograms.

V.A. Efficiency and Time-of-Flight Parameters

Several intrinsic parameters are common to all geometries. Two of these are the Thickness (mm) and the Efficiency that is defined by the detection probability for 1-Å neutrons passing perpendicularly through the thickness. The thickness represents both the physical dimension of the detector element, and also the active volume. The active detector material is assumed to have $1/v$ absorption; e.g., ^3He , ^6Li , or ^{10}B . If the efficiency is set to unity, all wavelengths will be detected at the first surface of the detector and none will be transmitted. For a detector used as a beam monitor, you may either set the 1-Å efficiency to a realistic low number (e.g., 10^{-4}), or set the “Special Feature: Monitor detector” to make the efficiency artificially unity with *no* absorption.

The other Special Feature is a magic “Polarization detector” that records separate histograms for each component of the neutron polarization when it reaches the detector.

V.A.1. Logarithmic time scale

For the majority of instruments at a pulsed source the most efficient time-of-flight encoding is logarithmic. Bin widths are thus represented by a ratio dt/t rather than a fixed increment Δt , so the *relative* wavelength resolution is constant. To generate a logarithmic scale, parameters Tmin (μs), $dt/t > 0$, and either Tmax (μs) or “# of Time Slices” are required (if the number of slices is entered, Tmax is computed).

If the Clock Tick (μs) is zero, the log scale is perfect to the limits of the computer. Practical clocks have a digital granularity that does not allow such a perfect logarithmic scale. Clock Tick (μs) is the period of the underlying master clock, leading to quantization of the bin boundary increments. For the Fastbus clocks at the Lujan center, the tick is either 0.1 μs or 0.05 μs .

Generally the Delay (μs) may be zero; this parameter was included in the data format to account for delays in the encoding electronics and to allow for calibration of the wavelength scale. After the logarithmic time bins have been computed, the value of Delay (μs) is *subtracted* from each bin boundary to generate the final detector time-of-flight scale. For some special cases (such as direct-geometry inelastic scattering) this parameter may be used to generate non-linear bins with optimized resolution. If the secondary flight path is time-focused with flight time T2, then setting the delay to $-T2$ will result in a time scale that is logarithmic in the incident flight time T1. As a more general example, suppose you want a counting frame from 30,000 to 50,000 μs , with bin width Δt varying from 3 μs at the start of the frame to 50 μs at the end. Consider a logarithmic scale with times at the beginning and end of T1 and T1 + 20000 μs , and find T1 such that the ratio of end/beginning times will be 50/3; this gives T1 = 1276 μs . Set up a log scale from Tmin = 1276 μs to Tmax = 21276 μs . A value of $dt/t = 0.00235$ will generate the required bin widths from 3 μs at Tmin to 50 μs at Tmax. Then set Delay = $-28724 \mu\text{s}$ so that the start/end times are as desired.

You should always enter a Frame frequency (Hz) for a pulsed source so that neutrons arriving in the “wrong” frame will be recorded as background. Generally this is the same as the source frequency, but

may be a submultiple if choppers are used to remove some pulses.

V.A.2. Linear time scale

A linear time scale will be generated when $dt/t = 0$. The required parameters are T_{min} (μs), T_{max} (μs), and “# of Time Slices.” You should always enter a Frame frequency (Hz) for a pulsed source so that neutrons arriving in the “wrong” frame will be recorded as background. Generally this is the same as the source frequency, but may be a submultiple if choppers are used to remove some pulses.

V.A.3. No time scale

If the “# of Time Slices” is 0 (or if there is not enough information to generate a time scale), then wavelength histograms will be recorded instead of time-of-flight. No control is provided for the wavelength bins; the scale is logarithmic with 1500 bins from 0.2 to 63 Å.

V.B. Plane Detectors

The extrinsic parameters define the center of the entrance surface of the box containing the detector; this is also the active surface of the detector. If Tilt and Slope (*cf.* Fig. 3) are 0, then the surface normal of the detector is parallel to the Z-axis. If the detector is offset in the +X-direction from the sample, as shown in Fig. 8, then it requires a *negative* Tilt if it is to face the sample. A detector at a back angle (detector Z less than sample Z) may be made to have its active surface face the sample by giving it a negative thickness, *or* by setting the Tilt to 180°. For a detector above the horizontal plane ($Y > 0$), Slope should be *positive*.

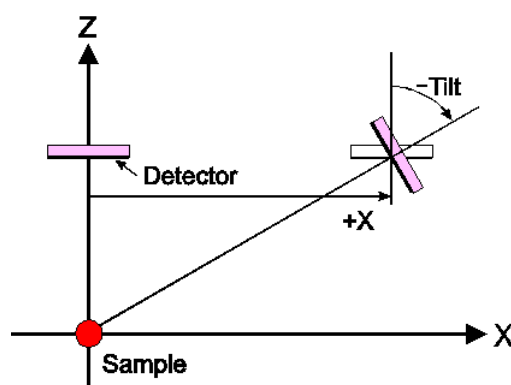


Figure 8. Tilt for a plane detector

Either shape of box may contain a detector to be encoded in either rectilinear or plane-polar coordinates; the choice is an intrinsic option. For rectilinear coordinates, parameters for “X” refer to the horizontal direction on the detector and “Y” is vertical (as seen on the detector face before Tilt and Slope are applied). The active area of the detector is not necessarily the same as the outline of its box, nor need it be symmetric. Most often X_{min} (mm) and Y_{min} (mm) will be negative, and X_{max} (mm) and Y_{max} (mm) will be positive, but it is also allowed to define the detector encoding in the negative direction. The values for “# X pixels” and “# Y pixels” must be > 1 for the detector to be encoded in the respective dimensions. For example, if “# X pixels” > 1 and “# Y pixels” = 1 (or 0), then the detector is a horizontal linear detector. If both are 1 (or 0) there is no position encoding, and if both > 1 the detector is two-dimensional. Since the electronics of position-sensitive detectors have uncertainties in addition to the contribution from pixel size, parameters are also provided for rms Encoding error (mm) in each direction.

Plane-polar coordinate use all of the same parameters, but “X” refers to the radial direction (mm), and “Y” refers to azimuth ($^{\circ}$). The detector active area is either a circle or an annulus (if $X_{min} > 0$), centered on the extrinsic center of the box. The azimuth is measured counter-clockwise from the +X-axis. Leaving both Y_{min} and $Y_{max} = 0^{\circ}$ defaults to full circle.

V.C. Cylindrical or Conical Detectors

For Detector (Conical Surface), the extrinsic position is a point on the axis of the cone (or cylinder), at the *midpoint* of the length of the detector box. This point is also the origin of the cylindrical coordinate system used to encode the detector. The symmetry axis must be parallel to one of the three coordinate axes, as chosen by an intrinsic parameter (*i.e.*, there is no extrinsic Tilt or Slope).

Length (mm), ϕ_{min} ($^{\circ}$), and ϕ_{max} ($^{\circ}$) describe the box containing the detector; Radius (mm) and

Slope (dimensionless) define the active surface. Length is measured parallel to the detector surface, and the azimuth angles ϕ are measured in the plane perpendicular to the chosen axis. For “X”, the azimuth is measured upward from the forward direction (from +Z to +Y); for “Y”, from the right side toward the forward direction (+X to +Z); and for “Z”, upward from the right-hand horizontal (+X to +Y), which is a left-handed system. The radius is measured at the mid-length of the detector. For a cylinder, Slope = 0. A positive value of Slope means that the radius increases in the + direction of the chosen axis.

The sign of the Thickness is significant. A positive value means that the “back” surface of the detector is at a larger radius, so the active surface is on the inside. For detection on the outer surface of the cone or cylinder, Thickness must be negative.

Encoding parameters for “X” are linear in the direction of the chosen axis of symmetry measured from the extrinsic center of the detector, and those for “ ϕ ” are for azimuthal angle as defined above. They may define smaller limits than the box dimensions. Most often Xmin (mm) will be negative and Xmax (mm) will be positive, but it is also allowed to define the detector encoding in the negative direction. The values for “# X pixels” and “# ϕ pixels” must be > 1 for the detector to be encoded in the respective dimensions. The resulting histograms may have 0, 1, or 2 physical dimensions in addition to the time-of-flight (or wavelength) dimension. Since the electronics of position-sensitive detectors have uncertainties in addition to the contribution from pixel size, parameters are also provided for rms Encoding error (mm or ° respectively) in each dimension.

V.D. Spherical Detectors

For Detector (Spherical Surface), the extrinsic position is the center of the sphere, which is also the origin of the polar coordinate system used to encode the detector. The polar axis is always parallel to the Z-axis (*i.e.*, there is no extrinsic Tilt or Slope).

The active surface of the detector is at Radius (m), and the spherical box may be limited by planes perpendicular to each axis. (Note that the radius and the plane locations are in m, contrary to the usual use of mm for transverse dimensions.) As an example, consider the Z-axis., which usually holes for the direct through beam. There are three ways to compute Zmin and Zmax.

- 1) You may already know the locations of the planes.
- 2) Given the radii r_1 and r_2 of the entrance and exit pipes and radius R of the detector surface,

$$Z_{min} = -\sqrt{(R^2 - r_1^2)} \text{ and } Z_{max} = +\sqrt{(R^2 - r_2^2)}.$$
- 3) Given the polar angles θ_{min} and θ_{max} , then $Z_{min} = R \cos(\theta_{max})$ and $Z_{max} = R \cos(\theta_{min})$.

If Zmin and Zmax are left at the default values of 0 (or if Zmin = Zmax), there are no bounding planes in the Z-direction. Similar methods can be used for the X- and Y-limits, if any.

The sign of the Thickness (mm) is significant. A positive value means that the “back” surface of the detector is at a larger radius, so the active surface is on the inside. For detection on the outer surface of the sphere, Thickness must be negative.

Both the polar angle “ θ ” and the azimuthal angle “ ϕ ” are encoded in °. Azimuth is measured upward from the +X-axis. If limiting planes in either X or Y are used, the azimuth scale will have blank areas. That is, ϕ_{min} and ϕ_{max} must include the full possible range. The values for “# θ pixels” and “# ϕ pixels” must be > 1 for the detector to be encoded in the respective dimensions. The resulting histograms may have 0, 1, or 2 physical dimensions in addition to the time-of-flight (or wavelength) dimension. Since the electronics of position-sensitive detectors have uncertainties in addition to the contribution from pixel size, parameters are also provided for rms Encoding error (°) in each dimension.

VI. Other Element Types in NISP_Win

The other element types, listed in Table 3, will be described briefly; information is available through

online help (Sec. II above). Since “Materials” are essential to many types, they will be described first.

VI.A. Materials – How to Cheat

Several general and several specific materials are defined in MCLIB. These may be selected from a pull-down menu in NISP_Win whenever a region of an element may be filled with a material. The simplest types are “Void,” meaning the neutron is transmitted without attenuation, and “Absorber,” meaning the neutron cannot enter the region at all. All other available materials are listed in file `Material.txt`. The data for each material are the number of parameters, the type number (algorithm) in MCLIB, and the model parameters.

Material types are:

- 1.0 amorphous, unpolarized
- 1.1 amorphous, polarized
- 1.2 amorphous, including center-of-mass correction
- 2.n polycrystalline Al, diamond, Fe, Be
- 3.n hydrogenous materials from $S(\alpha,\beta)$ tables
- 4.0 supermirror, unpolarized
- 4.1 supermirror, polarized
- 7.0 nuclear resonances with spin = 0
- 7.1 nuclear resonances with spin $\neq 0$, polarizable

Amorphous materials are represented by real and imaginary scattering-length densities to find surface reflectivity, and macroscopic cross sections to determine bulk transmission. The low-energy term is given at $v = 2200$ m/s and is proportional to $1/v$. Polycrystalline materials have explicit Bragg edges, and temperature dependent terms. Hydrogenous $S(\alpha,\beta)$ tables are taken from MCNP-5. Supermirrors have a critical-angle multiplier and the reflectivity at that angle; *i.e.*, the reflectivity falls linearly from unity at the normal θ_c to the value $R(\theta_{max})$ at $\theta_{max} = \text{Multiplier} \times \theta_c$. All polarizable materials are taken to be 100% polarized in the presence of *any* magnetic field (this “on/off” nature may be modified in the future).

You may also cheat by selecting “Enter by hand” as the material type. This allows you to use algorithms from MCLIB that have not otherwise been incorporated in NISP_Win. You should have a text file available with the specific data, in the same format as the `material.txt` file: the number of numbers (not counting itself), the MCLIB algorithm type, and the additional parameters needed. You might be entering a magnetic field definition, an algorithm from MCNP, or just some numbers you need to get into the geometry file. When the geometry file is created, NISP_Win will pause and ask you to enter the parameters, which you may either type by hand or cut & paste. Of course you could also enter a name and values in the `material.txt` file and it will appear in the drop-down menu.

The ultimate “cheat” is to edit values in the parameter block at the end of the geometry file. Use the definitions in file `\NISP\Includes\mc_elmnt.inc` to identify the parameters of interest. The block for each region begins with the material or algorithm type number. For the complete structure of the geometry file, see [Documents: icans14_1998.doc](#) and [MCLIB_2009.pdf](#).

VI.B. “Passive” Elements

This group of elements includes Aperture (Cone), Aperture (Rectangular), Beamstop, Block, Cylinder, Pipe, and Slab/Drift. Apertures and Pipes have interior and wall (body) materials, while the

Table 3. Other element types

| |
|------------------------------|
| Slab/Drift |
| Cylinder |
| Block |
| Pipe |
| Aperture (Cone) |
| Aperture (Rectangle) |
| Beamstop |
| Chopper (X) |
| Chopper (Y) |
| Fermi Chopper |
| Gravity Focus |
| Guide (Straight) |
| Guide (Segmented/Curved) |
| Soller Collimator (Straight) |
| Soller Collimator (Radial) |
| Bender/Multichannel Guide |
| Mosaic Crystal |
| Mosaic Crystal (Spherical) |
| Special Surface |

other types are single materials. The circular Aperture may be tapered, and the Block may be tapered, tilted, sloped, and rotated. The geometry of the simpler types may be made more versatile in the future. A Beamstop is required only if there is a detector in the direct beam, as in small-angle scattering. It has a feature that it is assumed to be removed for transmitted neutrons and in place for scattered neutrons; *i.e.*, a simultaneous transmission measurement is simulated.

VI.C. Choppers and Gravity Focuser

A simple model is used for disk choppers Chopper (X) and Chopper (Y), which have blades crossing the beam horizontally and vertically respectively. The extrinsic Z (m) is the upstream surface of the disk. The center of the rotating disk is located by the intrinsic parameter "Offset from the beam line" (mm) and the velocity of the edges is determined from that radius and the Rotation Frequency (Hz). Thickness (mm) is used both as the outer dimension of the element and as the thickness of the blade. The illuminated Width (mm) and Height (mm) should approximate the size of the beam. The signs of both are significant: the sign of Width for Chopper (X) (or Height for Chopper (Y)) is the direction of motion of the chopper crossing the beam. If the sign of the other dimension is negative, the chopper will have two counter-rotating blades instead of a single disk. Approximations of the model are first, the edges of the openings in the disks are assumed to move linearly across the beam, and second, all timing is measured at the first (upstream) surface of the disk. Phase and duration of the opening are expressed as times relative to the nominal zero time when the chopper is half-Open (μs) and half-closed (μs). If the disk is thick enough that the time of transit of the neutron at the nominal velocity is significant, the closing time should be made earlier and the Phase Jitter (μs) should be increased to emulate the "fuzziness" of the edge. However, in this model every neutron will either pass through a full thickness of void or of blade material.

A Fermi Chopper has a body that is a vertical cylinder with a central opening containing a set of vertical curved collimating blades. The combination of high rotation speed and blade curvature acts both as a pulse generator and a narrow velocity selector. The extrinsic position is on the axis of rotation, at the mid-height of the slit package. Because of the sagitta of the blades, X should be offset by $d^2/32R$, where d is the rotor diameter (mm) and $(1/R)$ is the blade Curvature ($/\text{m}$), so that the position of the opening will be centered on the beam axis. For a neutron sufficiently fast that the path through the chopper can be approximated as circular, the relationship between curvature, Frequency (Hz), and the nominal neutron velocity is: $v_0 = 4\pi Rf$. Use this velocity to compute the desired "Nominal time-of-flight from source," (μs), which (in the case of a pulsed source) should include an appropriate delay for the source to peak following the start of the proton pulse. Leave both parameters defining the macroscopic cross section (cm^{-1}) of the rotor body 0 to ignore absorption of neutrons in the rotor itself. "Thermal" means that part of the cross section proportional to $1/v$, evaluated at neutron velocity 2200 m/s. Typical values (for aluminum) are: high-energy = 0.085 cm^{-1} , thermal = 0.017 cm^{-1} . If values are entered for the body, then values are also *required* for the blade material cross section. For a composite blade that is 50% B_4C , typical values would be: high-energy = 0.30 cm^{-1} , thermal = 42.19 cm^{-1} . If all four of these entries are 0, then the body is transparent and the blades are perfect absorbers. *Note: the current algorithm for partial transmission is approximate.* The "Number of blades" is one more than the number of channels between blades.

A Gravity Focus element is a circular aperture which moves upward during each pulse so the parabolic trajectories of all wavelength neutrons can be brought to a common focus. The Frequency (Hz) should be the same as the pulse frequency. The motion is described by constant Acceleration ($\text{m}/\mu\text{s}^2$) between times T_{start} (μs) and T_{top} (μs), and linear recovery to the baseline for the next pulse. To compute the proper acceleration, let Z_1 be the distance from the source to the collimator entrance aperture, Z_2 the distance from the source to the gravity focuser, and Z_3 the distance from source to focus point. Then $a = (g/2) (Z_3/Z_2 - 1) (1 - Z_1/Z_2)$.

VI.D. Guides/Bender

Three forms of guides are available: Guide (Straight), Guide (Segmented/Curved), and Bender/Multichannel Guide. One difference is that only a straight guide may have Tilt and Slope. A straight guide may be tapered using intrinsic parameters Horizontal Divergence and Vertical Divergence (negative values for convergence, positive for divergence). The “curved” guide has a Total Bend Angle ($^{\circ}$, positive to the left) and a Number of Segments, which are identical straight segments. The Bender does have circular curvature, and may also be used as a single- or multi-channel curved guide.

The material in the interior of the guide may be Void or Air. The horizontal and vertical substrates are the same thickness and material, but the reflective coatings may be different. The horizontal and vertical Roughness parameters describe the long-range waviness of the corresponding surfaces. If > 0 , a randomly oriented vector of this length is added to the unit normal vector of the surface. The magnitude (if < 1) is the sine of the maximum angular deviation from the perfect surface. If the parameter is < 0 , the surface is completely random (not useful for guides!).

When a neutron exits from either form of curved guide, the Z-axis may have been rotated by the bend angle. Neutrons that penetrate the substrate before reaching the end of the guide are in an intermediate coordinate system and need to be transported to the exit surface; for this reason the “Exterior Material” must also be defined in the case of the curved/segmented guide.

VI.E. Mosaic Crystals

Mosaic crystals are available in two geometries, Mosaic Crystal (Flat) and Mosaic Crystal (Spherical). In the flat case, the extrinsic position is the center of the first face of a rectangular crystal with defined Width (mm) and Height (mm), and Tilt/Slope (Fig. 3) describe the surface normal pointing into the crystal. For the spherical case, the extrinsic position is the center of the sphere and Tilt/Slope refer to the polar axis of the sphere (default is the Z-axis). The size of the spherical crystal is given by limits of azimuthal and polar angles (all in $^{\circ}$). Thickness (mm) is a very important parameter for mosaic crystals, both in Bragg (reflection) and Laue (transmission) geometry.

The face of the crystal does not have to be cut parallel to the reflecting planes. Suppose the crystal (or element of the crystal, in the spherical case) is rotated to the Z-axis by applying the inverse of the Slope followed by the inverse of the Tilt. Let the vector τ be normal to the reflecting planes. The angle between τ and the Z-axis is χ ($^{\circ}$) and the projection of τ on the X-Y plane is rotated by an angle ψ ($^{\circ}$) from the horizontal. The components of τ are thus $\sin\chi \cos\psi$, $\sin\chi \sin\psi$, and $\cos\chi$. The horizontal and vertical mosaic spread distributions are also applied in the form of sines of Tilt and Slope angles (respectively) in this rotated coordinate system. The given parameters for mosaic spread are standard deviations of Gaussian distributions. The “Spread of lattice spacings” is relative, $\Delta d/d$. If positive, it is the standard deviation of a Gaussian distribution, and if negative it is the fwhm of a Lorentzian.

The algorithm uses multi-step transport through the crystal with an exact solution of the Darwin equations at each step. Bragg reflection is based on the kinematical model. The algorithm is described fully in [Documents: ApPhysA_74\(2002\)S1458.pdf](#). Effects included are Bragg scattering, incoherent and coherent-inelastic scattering, and nuclear absorption.

VI.F. Soller and Radial Collimators

Extrinsic parameters of type Soller Collimator are the position of the center of the entrance, and the Tilt/Slope of the axis of the collimator. An intrinsic parameter “Blade Orientation” chooses whether the blades are Vertical or Horizontal. (Vertical blades remain vertical for any tilt, slope, or divergence, but horizontal blades are affected by slope and divergence.) Width and Height of the package are measured at the entrance plane, and Divergence is the ratio of change in total collimator width (or height if horizontal) per unit length along the axis; that is, the tangent of the divergence angle. Negative values indicate convergence, and positive mean divergence. The number of openings in the Soller collimator is

Nslots. The number of blades will be one greater than this, but the outermost blades will be half-thickness and coated only on the inner side. Each blade is defined by a substrate material and thickness, and by the materials and thicknesses of the absorbing layers on either side. (The layer thicknesses may be set to 0 if the substrate is the absorber.) The Roughness parameters describe the long-range waviness of the corresponding surfaces. If > 0 , a randomly oriented vector of this length is added to the unit normal vector of the surface. The magnitude (if < 1) is the sine of the maximum angular deviation from the perfect surface. If the parameter is < 0 , the surface is completely random (quite useful for Sollers!).

Another form of collimator is Radial Collimator, designed for use between the sample and multiple detectors. The collimator is doughnut-shaped and is bounded by two concentric cylinders. The axis of the doughnut is the *negative* Z-axis rotated by the Tilt and Slope angles. The heights of bounding cylinders can be different. If they converge to 0 at the axis, the end caps of the collimator two sheets of a cone; otherwise they are planes. Blades are defined the same as for Sollers.

VI.G Using Special Surfaces to modify region geometry

Element type “Special Surface” can be used to define any form of quadratic surface. The connection matrix (sec. VII) can then be used to make the surface an additional boundary of the *first region* of one or more elements. For regions other than the first, it will be necessary to edit the Region Definition matrix in the output geometry. To visualize the modified regions, the .X3D file (sec. VIII) must be edited.

VII. How to Fill Space Between Elements

Neutrons are *not* restricted to flow sequentially from a given element to another in a list, but the space between all elements must be defined. In a linear (sequential) portion of the instrument, such as the incident flight path from source to the final aperture before the sample, drift sections between elements may either be defined explicitly or inserted by NISP_Win as “automatic” drifts. A simple complete instrument with a “flat” sample and a single detector in the forward direction may also be modeled as linear. In more complex spatial arrangements, especially the collection of elements following the sample, it becomes too difficult to define all of the space between elements as concave volumes. You will instead define a “Scattering Chamber” in which multiple elements are embedded. This form of connection allows arbitrary paths between elements, but transport is somewhat slower because *every* surface of *every* enclosed element must be tested every time a neutron moves. Note that stray neutrons following unexpected or roundabout paths are also tracked. Tracking such parasitic neutrons, which compromise the performance of an instrument, is a special feature of the NISP geometry.

To specify connectivity, click Connect > Enable. That will open a screen with a square matrix with a row and column for all defined elements, such as Fig. 9 that shows the first 19 elements of the LAPTRON instrument. Initially the matrix will be empty. You will make entries in it in by placing the cursor at the desired dot and either *left* click for a row-to-column sequential connection, or *right* click for a row-surrounded-by-column connection. To erase an entry, <Shift>-click on it.

After creating the connection matrix it is a good idea to save your instrument file. Click either File > Save or File > Save As. *There are no backups or automatic saves of the file.*

VII.A. Sequential Elements

To guarantee that neutrons at the output surface of one element will have a path to the input surface of another element, place an “X” at the dot connecting the output row to the input column, by positioning the cursor and *left*-clicking. Some elements do not have an output surface, and some don’t have an input surface. Those are indicated by hyphens instead of dots, and you can’t place an X there. At the time a geometry file is created, NISP_Win will insert “Drift” sections between sequential elements if needed. **WARNING:** do *not* use “X” if either of the two surfaces is *curved*, as overlapping regions may result. In Fig. 9, LAPTRON is connected to Bulk Shield , . . . , and Final Aperture is connected to Outer Chamber.

| | LAPTRON (f.p) | Bulk Shield | Pipe # 2 | Incident bea | Pipe # 3 | Final Apertu | Inner Chambe | Outer Chambe | Sample, CaF2 | Absorbing ch | Void chip [1 | North Anvil | West Anvil | South Anvil | East Anvil | Upper Anvil | Lower Anvil | Monitor @ 10 | Radiography | |
|--------------------------------|---------------|-------------|----------|--------------|----------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|------------|-------------|------------|-------------|-------------|--------------|-------------|---|
| LAPTRON (f.p.8) | - | X | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| Bulk Shield Pipe | - | . | X | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| Pipe # 2 | - | . | . | X | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| Incident beam shield | - | . | . | . | X | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| Pipe # 3 | - | . | . | . | . | X | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| Final Aperture | - | . | . | . | . | . | . | X | . | . | . | . | . | . | . | . | . | . | . | . |
| Inner Chamber (anvils, sample) | - | . | . | . | . | . | . | . | S | . | . | . | . | . | . | . | . | . | . | . |
| Outer Chamber (detectors) | - | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| Sample, CaF2 | - | . | . | . | . | . | . | . | S | . | . | . | . | . | . | . | . | . | . | . |
| Absorbing chip [10] | - | . | . | . | . | . | . | . | . | S | . | . | . | . | . | . | . | . | . | . |
| Void chip [11] | - | . | . | . | . | . | . | . | . | . | S | . | . | . | . | . | . | . | . | . |
| North Anvil | - | . | . | . | . | . | . | . | . | . | . | S | . | . | . | . | . | . | . | . |
| West Anvil | - | . | . | . | . | . | . | . | . | . | . | . | S | . | . | . | . | . | . | . |
| South Anvil | - | . | . | . | . | . | . | . | . | . | . | . | . | S | . | . | . | . | . | . |
| East Anvil | - | . | . | . | . | . | . | . | . | . | . | . | . | . | S | . | . | . | . | . |
| Upper Anvil | - | . | . | . | . | . | . | . | . | . | . | . | . | . | . | S | . | . | . | . |
| Lower Anvil | - | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | S | . | . | . |
| Monitor @ 10.55 m | - | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | S | . | . |
| Radiography Detector | - | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | S |

Figure 9. Connection matrix, for a portion of the LAPTRON instrument. An “X” (entered by a left click) indicates that neutrons must be able to pass from the element listed at the left to the element listed at the top, that is, from row to column. An “S” (right click) indicates that the row element is surrounded by the column element. The complete instrument has another 14 detector banks enclosed in the Outer Chamber.

VII.B. Use of a Scattering Chamber

Elements without input or output surfaces and elements not connected in a linear fashion must be enclosed within a “Scattering Chamber” (or within some other element). This is analogous to the way instruments are constructed, with the sample, secondary collimators, and detectors enclosed in a chamber. Three shapes of scattering chamber are provided. For instance, a Pipe may be used to enclose a guide.

VII.B.1. Chamber (Cylinder)

A common shape used for a scattering chamber is a vertical cylinder with its axis through the sample. Extrinsic parameters X (mm) and Z (m) locate the axis, and Y (mm) is the vertical position if the incident beam. The body of the chamber has a Radius (m) and incremental heights below and above the beam line; note that ΔY_Bottom (mm) must be *negative* and ΔY_Top (mm) must be *positive* for the beam line to enter the chamber! The incident beam pipe may protrude into the chamber in the form of a horizontal cylinder. Its radius is given in mm, and the offset of the window between the beam line and the chamber is entered as ΔZ (m), which is usually negative because the beam pipe ends before the center of the chamber. Elements upstream from the window are part of the primary beam line and are not inside the chamber. The “material” in a scattering chamber is always void; *i.e.*, neutrons drift freely between elements.

VII.B.2. Chamber (Long)

The secondary flight path may be enclosed in a horizontal pipe; this is typical of a small-angle scattering instrument. The extrinsic parameters define the center of the entrance face of the chamber. The Radius of the chamber and its Length are both given in m.

VII.B.3. Chamber (Sphere)

A *virtual* scattering chamber may be defined as a sphere centered at X (mm), Y (mm), Z (m), with a large enough Radius (m) to enclose all detectors and any other elements after the sample. The incident beam pipe may protrude into the spherical chamber in the form of a horizontal cylinder. Its radius is given in mm, and the offset of the window between the beam line and the chamber is entered as ΔZ (m), which is usually negative because the beam pipe ends before the center of the chamber. Elements upstream from the window are part of the primary beam line and are not inside the chamber. Remember to press the “Update” button at the bottom of the screen.

VII.B.4. Identify Elements in the Chamber - Nesting

Open the window with the Connection matrix (Fig. 9). A row element is embedded within a column element by *right-clicking* for the matrix element. In the example shown, the column for the Outer Chamber shows that it surrounds the Inner Chamber and a Beam Monitor (in addition, in the full instrument there are 14 more detector banks in the Outer Chamber). Then the Inner Chamber contains the Sample and several other elements. Finally, the Sample surrounds two radiography phantoms: a void and an absorber. This is an example of nesting elements. Even a solid element may have other elements embedded.

VIII. How to Visualize an Instrument

Making a picture of the instrument, such as Fig. 10, is a very important tool to see that the locations of the elements are what you intended. Visualization of instruments in NISP_Win is through a .X3D file, which is essentially a VRML (Virtual Reality Modeling Language) file converted to an XML syntax. The player of choice is Octaga 2.0, which is free for personal non-commercial use, and you should download it from www.octaga.com. For further information on VRML and X3D, visit <http://www.web3d.org/>.

Click 3D View. You may keep the default settings, or change the viewpoint to a fixed location, and/or change the background to a solid color. (The “Sky/Ground” setting helps show which way is up when you start moving around.) Finally, click Write X3D File. The example in Fig. 10 represents the IRIS backscattering spectrometer at ISIS (with the analyzers increased in height by a factor of 5 to show the spherical shape). One lesson learned from the simulation was that there is crosstalk between the two halves of the instrument, which can be shielded by the blue cylinder behind the detectors in Fig. 10. The Octaga Player allows you to walk through the instrument, and to view from any angle. It is possible – but challenging – to start at the source and move through the insides of the guides to the sample.

IX. How to Create Magnetic Fields

When a polarized neutron moves in a magnetic field, it undergoes spin precession. We have developed a precise and fast procedure for integrating Bloch’s equation ([Documents: NIMA_457\(2001\)338.PDF](#)), and magnetic field definitions have been added to MCLIB.

Magnetic induction field (**B**) regions occupy the same space in world coordinates as material regions, but are not connected to them in any way. They can not be coupled in the connection matrix, and they do not appear in the X3D visualization. (Current loops, solenoids, and wires do appear, but not the boundaries of their regions.) They are allowed to have surfaces that are the same as material regions, but the surface-type tag in the Region structure will be ± 10 , which is the same as 0 (not-a-boundary) for

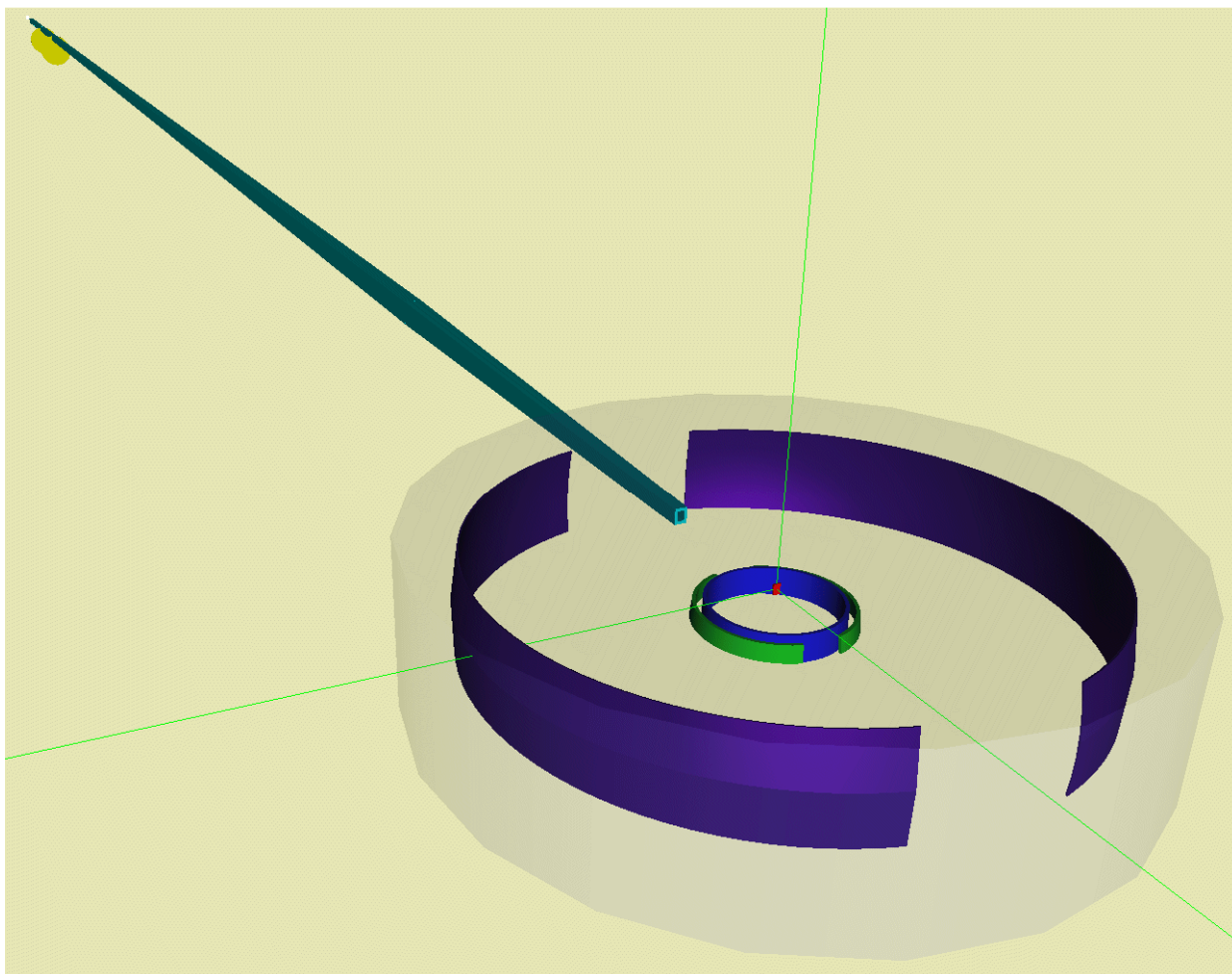


Figure 10. Visualization of a backscattering crystal-analyzer instrument. Elements are a source (white, barely visible in the distance), 2 Choppers, straight guides, a converging guide, sample chamber, sample (red cylinder), 2 mosaic crystals (spherical), 2 conical detector banks (green), and a shielding pipe (blue) to prevent crosstalk.

neutron transport. Most element types will include the definition of a cylindrical volume in which the field will be calculated. If you define overlapping regions, the resulting \mathbf{B} will be the vector sum. Magnetic regions do not have to fill all space.

A common feature of the various field types is that the fields may be pulsed in synchronization with the neutron source, such that $|\mathbf{B}|$ will be approximately proportional to neutron velocity over a specified time range. Of the various algorithms in MCLIB, four are implemented in NISP_Win: Magnetic Drift, Current Loops, Solenoid, and two Wire element types.

IX.A. Fields Specified with Fixed \mathbf{B} and Interpolation

Vectors \mathbf{B}_1 and \mathbf{B}_2 are defined to be constant on two planes perpendicular to the Z-axis, respectively at the extrinsic position Z (m) and at $Z + L$. For a “Uniform” field, \mathbf{B}_2 is ignored and $\mathbf{B} = \mathbf{B}_1$ anywhere between the two planes. You can choose from two forms of interpolation: “Linear” means that each component is interpolated, and “Helical” means that the magnitude and the azimuthal orientation of the vector are interpolated. Preceding and/or following fringe-field distances may also be given, in which case the field will ramp to zero in the respective distances.

IX.B. Fields Generated by a Number of Current Loops, or Wire Segments

There are two types to use arrays of loops or straight wire segments to generate \mathbf{B} . The field will be calculated anywhere within a cylinder, and will be zero outside those limits. For the “Current Loop” type, from 1 to 20 current loops (or point dipoles) may be defined, in **world** coordinates. For “Wire Segments,” up to 20 nodes are entered, and wires are drawn between those points in **world** coordinates. To close the loop you must enter a final node equal to the starting node. If more than 19 wires are needed, you can enter a second instance of the type.

IX.C. One or Two Rectangular Loops

A special case that uses the wire algorithm is two identical parallel flat rectangular loops (or a single loop). Extrinsic parameters X (mm), Y (mm), Z (m), Tilt ($^\circ$), Slope ($^\circ$), and Rotate ($^\circ$) specify the geometric center of the pair, the orientation of the long axis, and the rotation of the normal to the loops. The current in the two loops may be parallel or anti-parallel.

IX.D. Field of a Solenoid

Extrinsic parameters X (mm), Y (mm), Z (m), Tilt ($^\circ$), and Slope ($^\circ$) specify the center and the axis of the solenoid. The field will be calculated within a cylinder of specified length and radius, and will be zero elsewhere. The intrinsic parameters of the solenoid are its length, radius, and Ampere-turns. We use a complete analytic form for \mathbf{B} in the vicinity of a solenoid, involving complete elliptic integrals of the 1st, 2nd, and 3rd kinds ([Documents: icans16_2003.PDF](#)). A vector field plot is shown in Fig. 11.

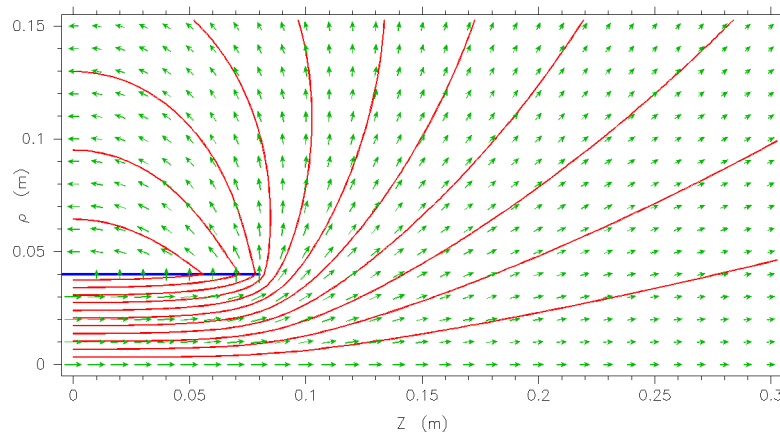


Figure 11. Magnetic induction from a solenoid. Red curves are contours of the vector potential, and the green vectors are our analytic solution for \mathbf{B} . (The magnitudes of the vectors are logarithmic, such that a length of $\frac{1}{4}$ of the vector at the origin is a factor of 10.)

X. How to Write Geometry Files

After all elements have been defined and the connection matrix updated, you should save your instrument file. Click either File > Save or File > Save As. There are *no* backups or automatic saves of the file.

To write the geometry file, click File > Geometry. NISP_Win will complete the process of building the instrument, including creation of drifts that may be needed between linearly connected elements, and loading parameters from specified file names. If all is well, a file with the name of the instrument file and extension `.geo` will be written.

The geometry file is formatted ASCII, and you may modify it with a text editor to adjust features of the instrument. Please read the description in the reference document if you wish to do this. The file will

have the current date-time-zone as part of its identification. To be user readable, the file also contains the complete `.inst` file, and can in fact be used as input to NISP_Win. [This is a suitable file to send to PASeeger@losalamos.com for debugging assistance!]

X.A. Multiple Files with Variable Parameters

To facilitate parametric studies, you can write as many as 35 geometry files with selected numeric parameters varying with linear or logarithmic steps between limits. You may choose as many as 40 variables. All of the selected variables change concurrently. The steps to do this are:

- 1) Click Elements > Vary Parameter, select an Element with a parameter to be varied, and click OK
- 2) Select an integer or real parameter from the list (*not* a file name, logical, or “list” variable). Interpolated integers will be rounded to the nearest integer.
- 3) Enter the First and Last values, and select whether interpolation is Linear or Logarithmic.
- 4) Set the number of cases, in the range [2,35] – whatever you select *last* will be the value used.
- 5) Repeat steps 1–4 for all desired variables, and then click File > Multiple Geom

The requested number of `.geo` files will be written. The names will be the instrument file name with an index character appended from the sequence 1–9, A–Z. The `.inst` information is *not* copied to each file, but the specific values of the variables are shown. The variable-parameter selections are not preserved in the instrument file.

XI. How to Execute MC_Run

When you have an executable for MC_Run (see the Appendix for installation instructions) and a geometry file you are ready to execute MC_Run. It is designed to run in interactive mode, but you may also construct a data file with the answers to the questions, and run in batch mode (see Appendix G).

XI.A. Initialization

The first question (first line of a command file) is the geometry file `[path]name`. If the file can be opened, the next question is a common name for all output files; by default the same name as the geometry file is used with different extensions (in particular, “.mc” and “.dat”). To accept the default, a carriage return may be entered (or a blank line in a command file). After the geometry file is read, *if* there is enough memory space available the user will be given the choice to have 3-D histograms (*e.g.*, X, Y, and t) for any 2-D detectors in the instrument. (In a command file, have a blank line for each 2-D detector.) All of the text from the beginning of the geometry file will scroll past (and be written on the `.mc` file), and if there are no errors in the file itself a test particle will be sent along the instrument axis as a simple sanity check. From this listing you can identify the internal numbers representing the entrance surfaces of regions along the beam line, which will help you choose surfaces to be monitored.

The default values of run-time parameters are defined in a Fortran namelist with the name “RUN.” The variables in the namelist are shown in Table 4, copied directly from the screen during execution of MC_Run. To change any values, enter “ \$RUN” (starting in column 2), then any number of entries of the form “NAME = value,” and terminate the list of changes with a “\$” or another line with a “\$” in column 2. Only parameters being changed need to be entered. To accept all defaults, enter “ \$RUN \$”.

Several parameters in the namelist control optional output files. If the user gives a MON_SURF number (the sequential number of the surface definition in the geometry file), an output “.mon” file will be written with every neutron that crosses that surface; if MON_SURF is negative, the corresponding source neutron will be recorded instead. One use of a `.mon` file is a source for subsequent runs (sec. III.D.3 above). If a value is *also* given for CORR_SURF, a correlation (`.cor`) file with the crossing of some preceding surface is written, with records in one-to-one correspondence with the `.mon` file. Setting

Table 4. Default Run-time Parameters for MC_Run

| | |
|-------------------|---|
| \$RUN | |
| MON_SURF = 0 | Surface number to monitor ("0" for none, negative for corresponding source neutron only) |
| CORR_SURF = 0 | "0" for no correlation, "-1" for preceding surface, " 1" for source, or "nnn" for surface nnn |
| SCAT = 'N' | Record all detected neutrons in files ('Y' for yes) |
| BAD = 'N' | Record complete traceback of Bad neutrons ('Y' for yes) |
| NSPLIT = 1 | Neutrons reaching sample may be used more than once |
| REUSE = 1 | Allowed number of times to reuse input source file |
| WT_MIN = 1.0E-06 | Minimum value of statistical weight to track |
| WT_FRAC = 1.0E-03 | Fraction of histories below WT_MIN to keep |
| SPEC = 'T' | Spectrum to be 'I'ncident or 'T'ransmitted? |
| POIS = 'N' | Integer output with Poisson statistics ('Y' for yes) |
| FIELD = 'N' | Include magnetic fields ('Y' for yes) |
| WRITEP = 'N' | Write Polarization after every move ('Y' for yes) |
| TRACE = 'N' | Plot trajectories ('Y' for yes). |
| GRAVITY = 'Y' | Include gravity ('N' for no). |
| DUMPS = 'N' | Keep ALL intermediate dumps ('Y' for yes). |
| ISEED = 12345678 | Random-number seed, 0 to randomize |

SCAT='Y' will generate “.sca” and “.tra” files with all neutrons detected in scattering and transmission mode respectively. A “.bad” file may also be generated which will contain complete (traceback) histories of any neutrons which are flagged as bad by some program module.

Neutrons reaching the sample may be split and scattered multiple times by giving a value to NSPLIT. This is an example of the variance reduction techniques used in MC_Run; for further discussion, see chapter 3 of J.Spanier and E.M. Gelbard, Monte Carlo Principles and Neutron Transport Problems, (Addison-Wesley, London, 1969). *Note: this should only be used if there are enough stochastic events downstream to limit the possibility of correlations in the data.*

The REUSE parameter applies only when a monitor file is being used as the input source. After the entire file has been read, it may be rewound and reread this number of times, after which the run will stop even if the requested number of neutrons has not been reached.

The user is also given control of the cutoff statistical weight WT_MIN below which a neutron is not tracked. These low-probability histories are not ignored completely. If WT_FRAC > 0, then each such neutron has a chance of WT_FRAC to have its weight multiplied by 1/WT_FRAC and to continue. To track more of the improbable events, decrease WT_MIN.

The user chooses whether downstream monitor spectra are to represent a sample-*in* ('T') or sample-*out* ('I') measurements. An option (POIS='Y') is also given to convert the final output histograms (which are sums of statistical weights) to integers, so that the error bars on the output will be Poisson distributed to look more like real data. If the statistical error from the number of histories is small, then the value for each bin will be replaced by a sample from a Poisson distribution with mean equal to the tally for that bin. If the error bars are already appropriate, then a simple Russian roulette procedure is to add a random number (on the range 0–1) and truncate the result to an integer.

If any magnetic field regions are present, the default for FIELDS will be 'Y'. These can be turned off by setting FIELDS='N', in which case polarization and neutron spin precession will not be calculated and execution will be significantly faster. Setting WRITEP='Y' will generate a “.pol” file with a record of the polarization vector (high precision) after every move in a magnetic field region

The TRACE option is described in the next section. By default, gravity is always on and intermediate dumps are overwritten on the same file. The options can be changed by setting GRAVITY='N' or

DUMPS='Y'. (If you compile MC_Run in the debug mode, the defaults will be TRACE='Y' and DUMPS='Y'.)

To make statistically independent runs of the same problem, you must change the starting value of the random-number generator, ISEED. In batch mode the default for subsequent runs is the final random number of the preceding run. When submitting multiple runs on a network you should set ISEED=0, which will initialize using the system clock so that the separate runs will be statistically independent.

After entering the namelist changes, you will be asked for the number of histories to be detected. The Monte Carlo run then begins. It will run until the specified number of histories has been detected after scattering, *or* until the number of neutrons started is 1024 times the requested number.

XI.B. TRACE Mode

When beginning a new instrument simulation it is useful to see where the neutrons are going – especially if none are being detected! Setting TRACE='Y' will open an interactive window using PGPlot graphics. Fig. 12 is a sample trace for an instrument similar to TOSCA at ISIS. The trace is controlled by keystrokes, as shown in an onscreen message (covered by traces in Fig. 12!):

| | |
|-----------------------------------|---|
| C,+,- = Center,ZoomIn,ZoomOut | each + or - changes scale by a factor of 2 |
| R,L,U,D = Right,Left,Up,Down | |
| 1-9 = transverse*1-5,10,20,50,100 | the transverse direction can be magnified compared to Z |
| E,P = Elevation,Plan | switch between side view and top view |
| V,B,N,M = plot 10000,1000,100,10 | |
| X,Z = Exit,Continuous | no more keystrokes will be read |
| other = plot 1 | |

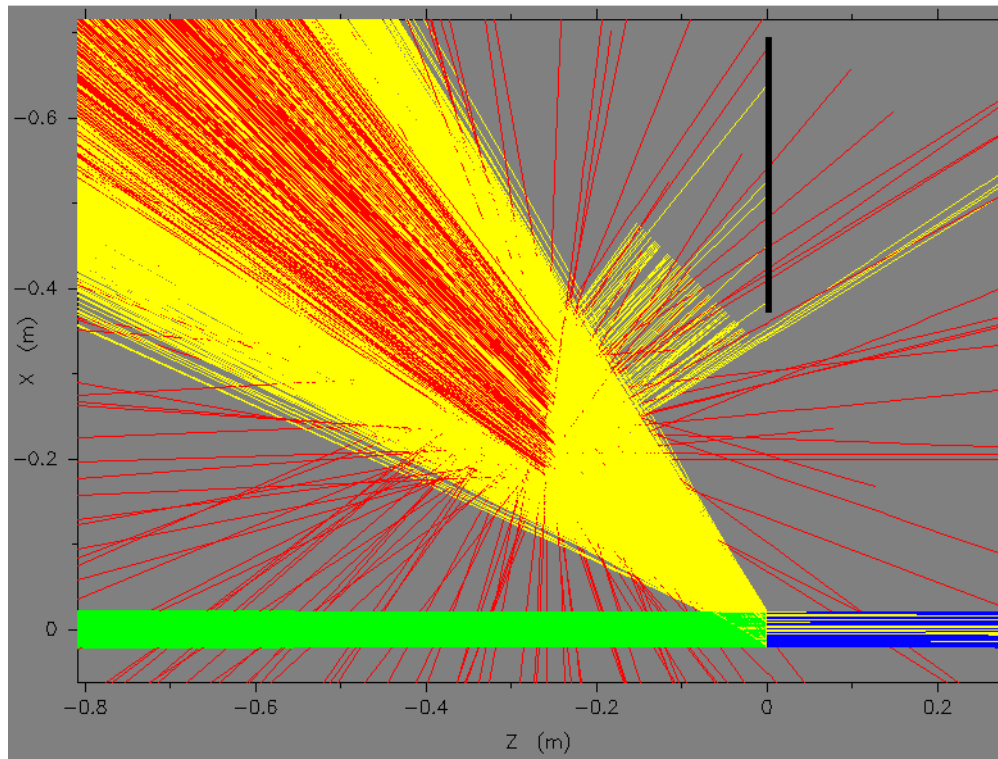


Figure 12. Neutron trace from MC_Run, showing trajectories at the sample, analyzer, filter, and detector (added by hand) of a TOSCA-style inelastic spectrometer. Green = incident beam, blue = transmitted, yellow = scattered, and red = “bad” neutrons either transmitted or non-Bragg scattered from the analyzer. The solid angle of scattering has been limited to illuminate the analyzer (see Sec. IV.E).

For instance, place the cursor at the point of interest and type + to enlarge, type B to add 1000 more traces to the picture, type C to clear the traces and start over, or type X to quit tracing and let the run go.

XI.C. Intermediate Output

During the run an occasional line is written on the screen showing where a neutron died. The line shows the source number, the number detected (only if this neutron was detected), the position and time, and the region and surface numbers where the history was terminated. This output continues at decreasing frequency to let you know the computer is still running, but may also be used for simple diagnostics. For instance, if no neutrons are being detected but most histories are dying at the same surface, there may be a chopper phase error or a misalignment. A warning message will also appear if a neutron has taken an inordinate number of steps.

For long runs, intermediate `.dat` files are dumped after the first hour and every two hours thereafter, to protect against computer crashes. At each dump an estimate is also made of the remaining time to finish the run.

XI.D. Visualizing and Analyzing Output

Output from `MC_Run` always includes a summary file and a file with detector data, and possible monitor and debugging files. Two programs are provided as part of NISP: `See_MC_Data`, and `SuperKnow`. Other data reduction programs are available in a separate downloadable file.

XI.D.1. Statistics of the Monte Carlo Simulation

Output from `MC_Run` always includes a summary file and a file with detector data, and possible monitor and debugging files. Two programs are provided as part of NISP: `See_MC_Data`, and `SuperKnow`. Other data reduction programs are available in a separate downloadable file.

The `.mc` file contains everything that was written to the screen during the run, plus summary statistics. You should check the list called “Beam elements along the instrument axis” (located just before the start of periodic output) to see that it makes sense. Remember that statistical weighting is used for variance reduction. In the summary (at the end of the file), integers refer to a number of histories and values in exponential form are sums of statistical weights, proportional to the “true” number of neutrons. For instance the line

```
Total histories started:      1839906 (1.278E+08 neutrons)
```

means that the source was sampled 1,839,906 times, and the sum of the resulting statistical weights represented 1.278×10^8 neutrons. The summary shows what fraction of the starting statistical weight was absorbed in every region of the instrument. The value for the first region after the moderator surface (often an automatic drift) will include any neutrons with projected start points outside the moderator surface, and also any neutrons from the sampled spectrum with wavelengths outside the limits given in the source definition. Percentages for all other regions exclude the neutrons omitted at the source

Neutrons are detected in two modes, Transmission and Scatter. Every history is born in Transmission mode, but when any neutron hits a sample it is split into two (or more, if `NSPLIT > 1`) histories. The transmitted neutron continues with reduced statistical weight without interaction in the sample, and the remaining statistical weight is assigned to `NSPLIT` histories (in Scatter mode) that *do* interact in the sample. Counts in every detector are shown in both modes. (If there is no sample in the simulation, detected neutrons are counted both as transmitted and as scattered.) The centroid and standard deviation are shown for transmitted neutrons; thus you have a measure of beam location and spread even for a non-position-sensitive detector. If the source brightness is known from the spectrum definition, then these detected numbers will be normalized to source MW-s; otherwise they will be normalized to starting weight.

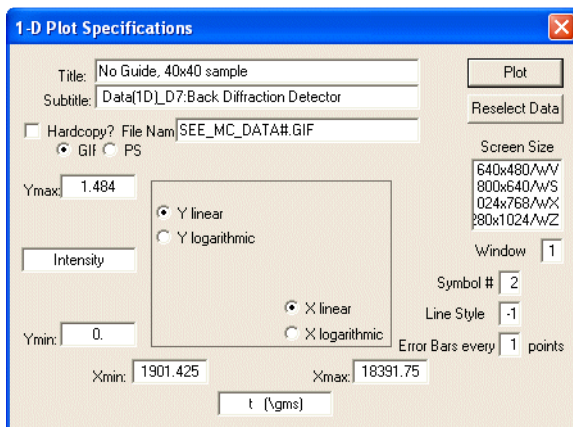


Figure 13. 1-D plot specifications in See_MC_Data. You may choose labels, log or linear, and plot limits. Default symbol is + and error bars are included on every point. To draw a solid line *only*, set Symbol # = -1, Line Style = 0, and Error Bars every 0 points. Screen Size is plot area in pixels; output files are always 900x1200.

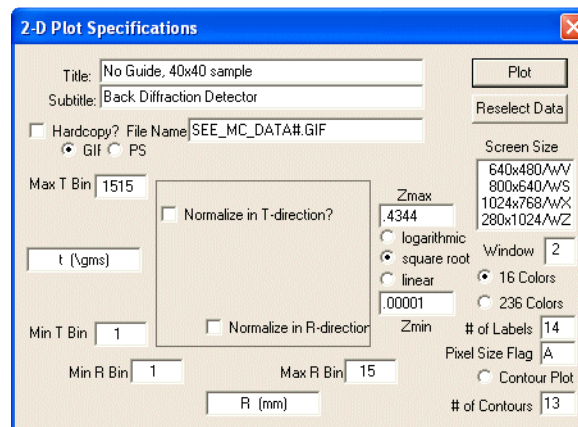


Figure 14. 2-D plot specifications in See_MC_Data. If the data file includes AREA blocks they may be used to normalize. Plot limits are integer bin numbers. Default is 16-color histogram with square-root scale. The Pixel Size Flag is the number of screen pixels per bin (1-9); “A” = aspect 5:3, “B” = biggest, “0” = largest squares.

Neutrons passing through choppers in the wrong phase (and some other “bad” conditions) are counted as “questionable neutrons passed.” If any of these neutrons actually reach a detector, they are counted as “questionable neutrons detected.” Having *any* neutrons in this category is a cause for concern.

XI.D.2. Simulated Data

The Windows application See_MC_Data has versatile plotting options to visualize all of the 1-D and 2-D detector data in the .dat file. It generates a Table of Contents and then uses the dialog boxes shown in Fig. 13-14 to set plot parameters. Plots are generated with PGPlot graphics; some special features for text are “g” for “next character Greek,” “A” for Å, and “d” or “u” to shift down or up respectively. We invite you to experiment with the options, and click Plot to see the results. Typing any key while the cursor is in the active plot screen returns you to the specification window. Plots may be written to files in either GIF or PostScript formats. Source code is provided, but this program is definitely *not* portable.

While the content of the .dat file represents a real measurement as detector histograms, a .mon or .cor file includes details that could not be measured, such as correlations between actual wavelength and the wavelength measured by time of flight. We refer to this as “super knowledge.” To extract this information from a .mon file (direct-access binary, *not* portable between systems), use program Super_Know with modifications for your specific needs. The template also shows how to extract correlated information between .mon and .cor files. Comments in the source code indicate where the user-specific code is to be inserted. Output files (.super) may be viewed with See_MC_Data.

In general, the user will be expected to use his/her own analysis routines, the same as if the data originated from a real instrument. This will generally require a conversion of formats from the standard format used in MC_Run ([Documents](#): DataFormat_1999.doc). Subroutines READ_1D and READ_2D in the MCLIB library may be used within your own Fortran code, or programs SWITCH and SPREAD can be downloaded from [Downloads](#): Analysis.zip. SWITCH will read a 1-D histogram and convert it to column format, and the program SPREAD will convert 2-D data to a spreadsheet format. For 3-D files, follow the example in sample program QLfromXYt. For other commercial graphics packages (IDL, SigmaPlot, etc.) the converted files from SWITCH and SPREAD will probably provide the most appropriate input.

Additional data reduction programs specific to certain classes of instrument are included on the Analysis.zip file. See Appendix.E for a list. Program DLfromZt is described in the Sec. XII.E.

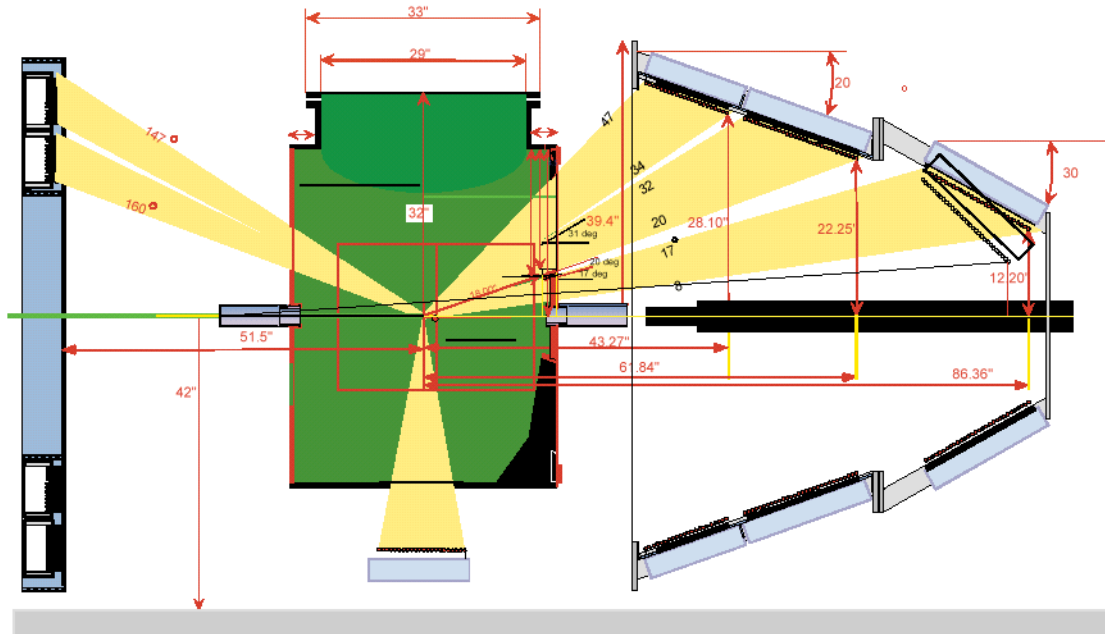


Figure 15. Sample chamber and detectors for HIPPO. For the tutorial, we will represent the six detector banks as annuli and cones (cf. Fig. 16). Dimensions are given in “inches”; 1 inch \equiv 25.4 mm.

XII. Tutorial Example: HIPPO

You can work through this relatively simple example, the High Intensity Powder and Preferred Orientation diffractometer (HIPPO) on f.p. 4 at LANSCE, and compare your results with ours, which can be found at [Downloads: HIPPO_data.zip](#). The first step is to sketch the layout; we used the design

Table 5. Parameters for “Source (Plane)”

| Parameter | Value |
|---|--------------------|
| Instrument Name | HIPPO, f.p. 4 |
| Center of Plane Source, X (mm) | 0 |
| Y (mm) | 0 |
| Z (m) | -9.00 |
| Horizontal Tilt of normal from Z-axis (°) | 0 |
| Source Area, Width (mm) | 130. |
| Height (mm) | 130. |
| Phase-space aperture 1: Distance (m) | 1 |
| Elevation (mm) | 0 |
| Diameter or Width (mm) | 106. |
| Height (if rectangular) (mm) | 0 |
| Phase-space aperture 2: Distance (m) | 8.25 |
| Elevation (mm) | 0 |
| Diameter or Width (mm) | 5.1 |
| Height (if rectangular) (mm) | 0 |
| Gravity-focus extension (mm) | 0 |
| Proton Pulse Frequency (Hz) | 30 |
| Proton Pulse Width (μ s) | 0.3 |
| Enter Source Name OR Select File | LujanII_04_H2O.tbl |
| Minimum Wavelength from file (\AA) | 0.25 |
| Maximum Wavelength from file (\AA) | 10. |

specification drawing in Fig. 15. Not shown in the diagram are the source and the T_0 chopper. Create a folder for the files, and in NISP_Win, click File > New, go to the folder, and enter File name HIPPO. Make the Title “HIPPO Tutorial.” Our model has 16 Elements: Source, 2 Pipes, T_0 Chopper, 2 Apertures, 2 Monitor detectors, Sample, 6 detector banks, and a Chamber.

XII.A. Use NISP_Win to Define the Beam Elements

[1] Elements > Add Element > Source (Plane) > OK (see Fig. 4 and Table 5). Name the Instrument “HIPPO, f.p. 4”. To eliminate the suffix “[1]” which is added to new elements to show the sequence number, re-edit the element and delete it from the name. This instrument is located on f.p. 4 at LANCE which is normal to

the moderator; use the “Lujan Mark II” table file. The source-to-sample distance is 9.00 m, and we define the origin of coordinates to be at the sample.

[2,3] Elements > Add Element > Pipe > OK. We represent the penetration of the bulk shield and the evacuated pipe from chopper to chamber as “Pipes.” The exact locations and lengths of the pipes are taken from layout sketches, accounting for the thicknesses of other elements. See Table 6.

Table 6. Parameters for “Pipe”

| Parameter | Pipe 1 | Pipe 2 |
|------------------------------|----------|----------|
| Element Name | Pipe 1 | Pipe 2 |
| Center Pipe entrance, X (mm) | 0 | 0 |
| Y(mm) | 0 | 0 |
| Z (m) | -7.996 | -2.700 |
| Horizontal (CCW) Tilt (°) | 0 | 0 |
| Vertical (upward) Slope (°) | 0 | 0 |
| Length (m) | 4.996 | 1.920 |
| Inside Radius (mm) | 100 | 35 |
| Wall Thickness (mm) | 10 | 3 |
| Roughness | 0 | 0 |
| Interior Material | void | void |
| Wall Material | absorber | Aluminum |

[4] Elements > Add Element > Chopper (X) > OK (see Sec. VI.C). See Table 7.

Table 7. Parameters for “Chopper (X)”

| Parameter | Value |
|---------------------------------|------------|
| Element Name | T0 Chopper |
| Location of Chopper disk, Z (m) | -3.0 |
| Disk Thickness (mm) | 300. |
| Vertical offset of axis (mm) | -250. |
| Beam area: Width (mm) | 50. |
| Height (mm) | 50. |
| Rotation Frequency (Hz) | 60. |
| Number of openings | 1 |
| Times when edge=0: Open (μs) | 500. |
| Closed (μs) | 16400. |
| Phase Jitter (μs) | 30. |
| Disk Material | absorber |

[5,6] Elements > Add Element > Aperture (Cone) > OK. Tapered or conical surfaces may be used for beam-defining apertures to reduce scattering; ideally, the surface is shielded either from the source or from the sample so there is no reflecting path. However, HIPPO uses cylindrical surfaces. The parameters of the apertures are given in Table 8.

Table 8. Parameters of Aperture (Cone)

| Parameter | Entrance | Exit |
|-----------------------------|-------------------|---------------|
| Element Name | Entrance Aperture | Exit Aperture |
| Aperture entrance, X (mm) | 0 | 0 |
| Y (mm) | 0 | 0 |
| Z (m) | -8.00 | -0.745 |
| Horizontal (CCW) Tilt (°) | 0 | 0 |
| Vertical (upward) Slope (°) | 0 | 0 |
| Thickness (mm) | 4. | 4. |
| Upstream Radius (mm) | 52.2 | 2.5 |
| Downstream Radius (mm) | 52.2 | 2.5 |
| Internal surface Roughness | 0 | 0 |
| Outer Diameter (mm) | 200. | 76. |
| Aperture Material | void | void |
| Body Material | B4C | B4C |

[7,8] Elements > Add Element > Detector (Plane Circular) > OK (see Sec. V). The first of the two monitor detectors has the “Monitor” flag set, giving it 100% efficiency with 0 absorption. The downstream monitor is treated more realistically, with efficiency (at 1 Å) = 85%. Parameters are in Table 9. The other two detectors in the table are described below.

Table 9. Parameters of Detector (Plane Circular)

| Parameter | Upstream | Downstream | 150° | 157° |
|------------------------------------|------------------|--------------------|-----------------------|-----------------------|
| Element Name | Upstream Monitor | Downstream Monitor | 150° Annular Detector | 157° Annular Detector |
| Center of front face, X (mm) | 0 | 0 | 0 | 0 |
| Y (mm) | 0 | 0 | 0 | 0 |
| Z (m) | -0.78 | 0.860 | -1.308 | -1.308 |
| Horizontal (CCW) Tilt (°) | 0 | 0 | 0 | 0 |
| Detector box Thickness (mm) | 10. | 10. | 100 | 100 |
| Outer Case Radius (mm) | 38. | 12. | 875 | 662 |
| Inner Radius (mm) | 0 | 0 | 663 | 450 |
| Efficiency at 1 Å | N/A | 0.85 | 0.85 | 0.85 |
| Start of counting time frame (μs) | 450. | 540. | 600 | 600 |
| End of counting time frame (μs) | 24000. | 28800. | 24000 | 24000 |
| Number of time channels | 0 | 0 | 10000 | 10000 |
| Logarithmic time resolution, dt/t | 0.01 | 0.01 | 0.0004 | 0.0004 |
| Clock tick (μs) | 0 | 0 | 0 | 0 |
| Log-scale delay (μs) | 0 | 0 | 0 | 0 |
| Frame Frequency (Hz) | 30 | 30 | 30 | 30 |
| 2D coordinates | N/A | N/A | Plane Polar | Plane Polar |
| Horizontal or Radial, Xmin (mm) | N/A | N/A | 849 | 626 |
| Xmax (mm) | N/A | N/A | 699 | 476 |
| # X pixels | 0 | 0 | 12 | 12 |
| rms X encoding error (mm) | N/A | N/A | 4. | 6. |
| Vertical or Azimuthal, Ymin (mm,°) | N/A | N/A | 0 | 0 |
| Ymax (mm,°) | N/A | N/A | 360 | 360 |
| # Y pixels | 0 | 0 | 1 | 1 |
| rms Y encoding error (mm,°) | N/A | N/A | 0 | 0 |
| Monitor detector? | yes | no | no | no |
| Polarization detector? | no | no | no | no |

[9] Elements > Add Element > Diffraction Sample (Cylinder) > OK (see Sec. IV.C). The calibration sample is Si powder. We use a cylindrical shell instead of a solid cylinder. See Table 10

Table 10. Diffraction Sample (Cylinder)

| Parameter | Value |
|------------------------------|------------------|
| Element Name | Si Powder Sample |
| Center of Sample, X (mm) | 0 |
| Y (mm) | 0 |
| Z (m) | 0 |
| Sample Outer Diameter (mm) | 10. |
| Inner Diameter (mm) | 8. |
| Height (mm) | 50. |
| Powder Sample File Name | Si.PDR |
| Fractional Density | 0.6 |
| Azimuthal angle, minimum (°) | 0 |
| Maximum (°) | 360 |

[10-13] Elements > Add Element > Detector (Conical Surface) > OK (see Sec. V). Parameters for the three conical detector banks at forward angles and the cylindrical bank at 90° are given in Table 11. We are leaving a 60° azimuthal gap at the top of the 90° bank to allow for the sample chamber. We need to include an estimate of the rms encoding error in the longitudinal dimension (X, mm) because the *real* detector banks are made from straight tubes instead of cone-shaped arcs. This table includes one extra parameter that will be needed for the data reduction program DLfromZt: distance from sample center to detector center. You will also need dp/dZ for each conical detector bank, which is “Slope” in Table 11.

Table 11. Parameters for Detector (Conical Surface)

| Parameter | 12° | 26° | 40° | 90° |
|---|----------------------|----------------------|----------------------|--------------------------|
| Element Name | 12° Conical Detector | 26° Conical Detector | 40° Conical Detector | 90° Cylindrical Detector |
| Center of coordinates, X (mm) | 0 | 0 | 0 | 0 |
| Y (mm) | 0 | 0 | 0 | 0 |
| Z (m) | 2.020 | 1.383 | 0.958 | 0 |
| Direction of Cone/Cylinder axis | Z | Z | Z | Z |
| Detector case, Length (mm) | 450 | 450 | 350 | 350 |
| minimum azimuth (°) | 0 | 0 | 0 | 120. |
| maximum azimuth (°) | 0 | 0 | 0 | 60. |
| Radius at midlength (mm) | 410. | 634. | 765. | 840. |
| Slope (0 if cylinder) | -0.577 | -0.364 | -0.364 | 0 |
| Thickness (mm) | 100 | 100 | 100 | 100 |
| Efficiency at 1 Å | 0.85 | 0.85 | 0.85 | 0.85 |
| Start of counting frame (μs) | 600. | 600. | 600. | 600. |
| End of counting frame (μs) | 24000. | 24000 | 24000. | 24000. |
| Number of time channels | 500 | 1000 | 2000 | 5000 |
| Logarithmic time resolution, dt/t | 0.008 | 0.004 | 0.002 | 0.0008 |
| Clock tick (μs) | 0 | 0 | 0 | 0 |
| Log-scale delay (μs) | 0 | 0 | 0 | 0 |
| Frame Frequency (Hz) | 30 | 30 | 30 | 30 |
| Longitudinal encoding, Xmin (mm) | 200. | 200. | 200. | 150. |
| Xmax (mm) | -200. | -200. | -200. | -150. |
| # X pixels | 32 | 32 | 32 | 24 |
| rms X encoding error (mm) | 7.5 | 4.5 | 3. | 0. |
| Azimuthal Angle encoding, Ymin (°) | 0 | 0 | 0 | 120. |
| Ymax (°) | 360 | 360 | 360 | 60. |
| # Y pixels | 1 | 1 | 1 | 1 |
| rms Y encoding error (°) | 0 | 0 | 0 | 0 |
| Monitor detector? | no | no | no | no |
| Polarization detector? | no | no | no | no |
| Sample-to-Detector distance, L ₂ (m) | 2.061 | 1.520 | 1.226 | 0.840 |

[14,15] Elements > Add Element > Detector (Plane Circular) > OK (see Sec. V). The two back-angle banks are annular and were included in Table 9 above. As for the forward detectors, rms encoding errors are entered for the radial direction because the actual detector elements are straight tubes.

[16] Elements > Add Element > Chamber (Long) > OK (see Sec. VII.B). The sample and all detectors must be in a virtual chamber to allow passage of neutrons anywhere. We choose a “Long” chamber, positioned to contain all of the detector banks. See Table 12.

Table 12. Parameters of Chamber (Long)

| Parameter | Value |
|--------------------|--------------------|
| Element Name | Scattering Chamber |
| Front face, X (mm) | 0 |
| Y (mm) | 0 |
| Z (m) | -1.50 |
| Length (m) | 4 |
| Radius (m) | 1 |

XII.B. “Connect” and Save Files

At any time while entering elements, you may save your instrument file by clicking File > Save. You may also generate a virtual reality view by clicking 3D View > Write X3D File. Then use an X3D player to examine the model. Fig. 16 is a sample view.

Open the connection window by clicking Connect > Enable. Since we did not define the elements in a particular order, you may have to check the Z-coordinates of the elements to see what follows what.

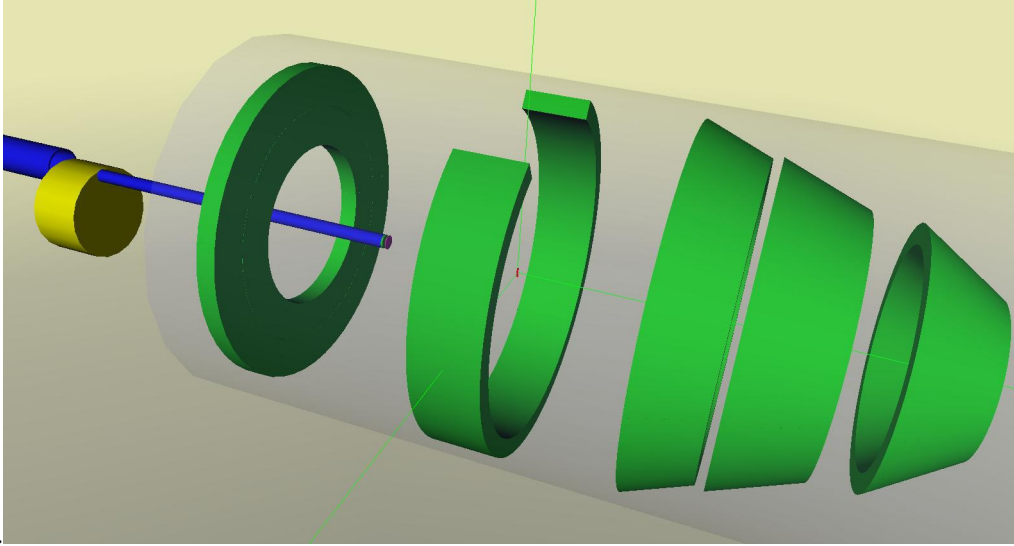


Figure 16. Virtual Reality rendition of the HIPPO model. The two pipes, the T0 chopper, and the six detector banks are easily seen. Also note the monitor detector and aperture at the end of the pipe, the sample (red cylinder) at the center, and the translucent scattering chamber. Comparing to Fig. 15, note that the 26° and 40° detectors have been modeled according to the *dimensions* on that drawing, which do *not* align as shown there.

The order should be Source → Entrance Aperture → Pipe 1 → T0 chopper → Pipe 2 → Upstream monitor. Those 5 row-to-column points in the Connect matrix should be marked with “X” by *left*-clicking. The Upstream Monitor and all of the remaining elements are totally enclosed within the Chamber, which is marked by *right*-clicking and the letter “S” in the matrix. To remove any Xs or Ss from the matrix, hold the <Shift> key and click. The final matrix should look like Fig. 17.

| | HIPPO, f.p. | Pipe 1 [2] | Pipe 2 [3] | T0 Chopper [4] | Entrance Aperture [5] | Exit Aperture [6] | Upstream Monitor [7] | Downstream Monitor [8] | Si Powder Sample [9] | 12° Conical [10] | 26° Conical [11] | 40° Conical [12] | 90° Cylindrical [13] | 150° Annular [14] | 157° Annular [15] | Scattering Chamber [16] |
|-------------------------------|-------------|------------|------------|----------------|-----------------------|-------------------|----------------------|------------------------|----------------------|------------------|------------------|------------------|----------------------|-------------------|-------------------|-------------------------|
| HIPPO, f.p. [4] | - | - | - | - | X | - | - | - | - | - | - | - | - | - | - | - |
| Pipe 1 [2] | - | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - |
| Pipe 2 [3] | - | - | - | - | - | - | X | - | - | - | - | - | - | - | - | - |
| T0 Chopper [4] | - | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Entrance Aperture [5] | - | X | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| Exit Aperture [6] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| Upstream Monitor [7] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| Downstream Monitor [8] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| Si Powder Sample [9] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| 12° Conical Detector [10] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| 26° Conical Detector [11] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| 40° Conical Detector [12] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| 90° Cylindrical Detector [13] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| 150° Annular Detector [14] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| 157° Annular Detector [15] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | S |
| Scattering Chamber [16] | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

Figure 17. Connection matrix for the HIPPO tutorial. See also Sec. VII and Fig. 9.

After connecting, File > Save so that the connection matrix will be saved. Then click File > Geometry. You should receive one information message (single *) for the creation of an automatic drift, and two Warning messages (**) concerning input and output from the Chamber. Always check the messages; in this case the in/out connections are not required for the Chamber so it is all right.

XII.C. Run the Simulation with MC_Run

Follow the instructions in Sec. XI.A to start the simulation. Give the path and file name for the geometry file, and a blank line to use the same path and name for the output files. Set runtime parameter

```
$run nsplit=3 /
```

and $1E6$ neutrons to be detected (or $1E5$ if you are in a hurry). You may notice that with this NSPLIT value the number detected is larger than the number of starts. This is probably ok because there are so many independent detectors in the model that correlations are unlikely. If all is going well, get a cup of coffee. The execution time for 10^6 detected neutrons in a 2.8-GHz Pentium-4 was 10 minutes. If you are running the same version of MC_Run executable and used the default ISEED, the final random number should be -1475417440; if so, everything will be identical. Otherwise, results should be *statistically* the same as what is found in the file [Downloads: HIPPO_data.zip](#). For instance:

```
Total histories started:      754584 (7.551E+05 neutrons)
Detector_D3:12° Conical Detector [10]
detected in Scatter Mode:      20510 (3.978E+01 n, 4.891E+03 n/(MW-s))
Detector_D4:26° Conical Detector [11]
detected in Scatter Mode:      98097 (1.950E+02 n, 2.398E+04 n/(MW-s))
Detector_D5:40° Conical Detector [12]
detected in Scatter Mode:     193573 (3.617E+02 n, 4.446E+04 n/(MW-s))
Detector_D6:90° Cylindrical Detector [13]
detected in Scatter Mode:     566877 (8.992E+02 n, 1.106E+05 n/(MW-s))
Detector_D7:150° Annular Detector [14]
detected in Scatter Mode:      64835 (1.142E+02 n, 1.403E+04 n/(MW-s))
Detector_D8:157° Annular Detector [15]
detected in Scatter Mode:      55706 (9.927E+01 n, 1.221E+04 n/(MW-s))
```

XII.D. View Raw Data with See_MC_Data – Fix Problems

Fig. 18 shows downstream monitor transmission measurement as a function of time of flight. Fig. 19 is the raw 2-D histogram for the 40° bank. The intensity scale is logarithmic, the lower limit is 10^{-4} times the upper, and the 236-color option is selected. So far so good – but there is trouble in some of the files. There are a few neutrons reaching the detectors in “transmission” mode – that is, neutrons reaching detectors because of scatter in apertures or pipe walls instead of the sample. These represent background, but the problem is that they occur in a single detector element and time bin, rather than being spread over all time and space. The result occasionally is an individual cell with orders-of-magnitude higher intensity than its neighbors – a very unphysical situation. The offending histories can be identified because they are detected in Transmission mode as well as in Scattered mode. There were 3 in the 90° bank, 4 in the 150° bank, and 7 in the 157° bank. Ten high-weight histories were identified and subtracted from the HIPPO.dat file (by hand) before the raw data plots (in the HIPPO_data.zip file) were made.

XII.E. Reduce Raw Detector Data to Histograms vs. d, DLfromZt

The .dat file contains at least 16 data sets, including 2-D position/TOF histograms for each of the six detector banks, and the downstream monitor spectrum that was recorded in the “Sample In” mode. The sample data reduction program DLfromZt will convert time to wavelength (λ) and detector position to scattering angle (2θ). After normalizing for the incident spectrum, the solid angle of each detector

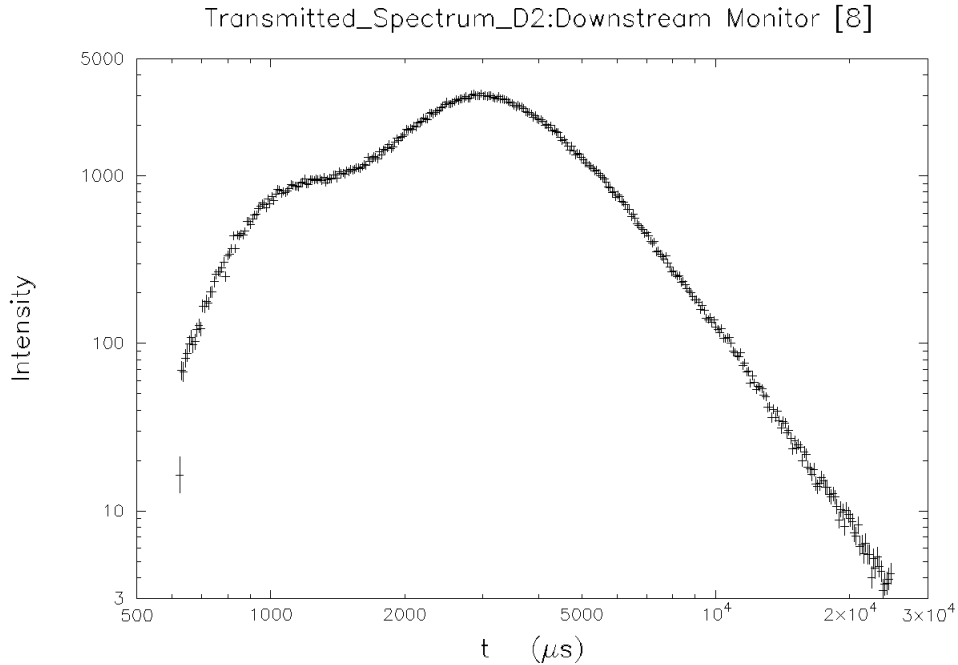


Figure 18. Time-of-flight spectrum. This is the beam intensity transmitted through the sample, and is used to normalize data in all detector banks.

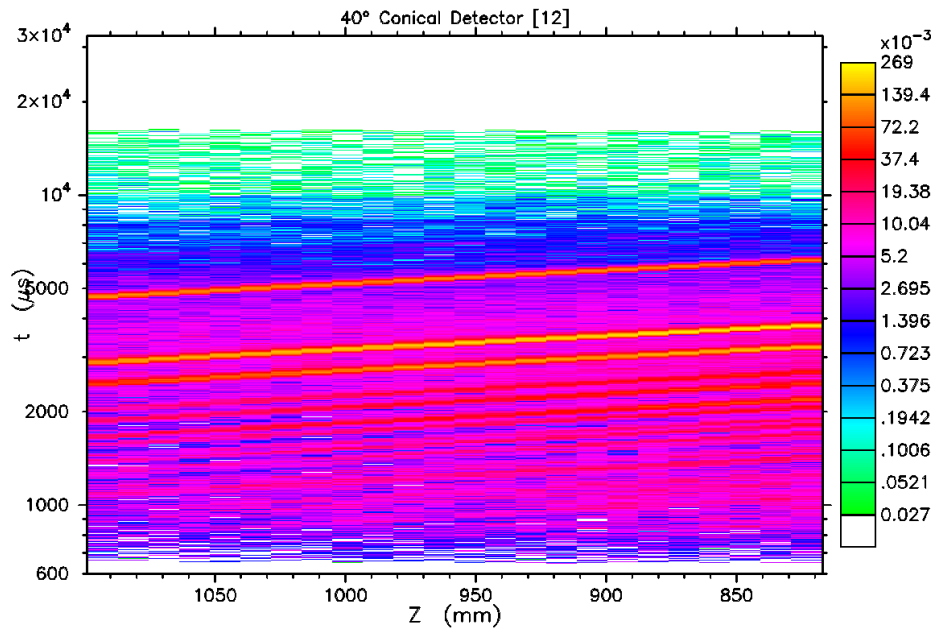


Figure 19. 2D detector histogram for the 40° bank. There are 24 elements in the Z-direction, which will convert to scattering angle 2θ (33.5° – 44.4°). The logarithmic time-of-flight scale is equivalent to wavelength λ (0.23\AA – 12.6\AA). The color intensity scale is logarithmic, covering four decades. The normalized summation of the data as a function of d-spacing is shown in Fig. 20.

element, and wavelength-dependent efficiency, the 2-D detector histograms are combined into logarithmic bins of $d = \lambda/(2 \sin\theta)$. Statistical errors are propagated for the intensity, and in addition the uncertainty in d is computed from the rebinning procedure. That is, both vertical and horizontal error bars are computed for the output data. An example is shown in Fig. 20.

The program is versatile in that it successfully deals with many different elastic scattering geometries. Unfortunately the versatility comes at the expense of requiring a lot of input from the

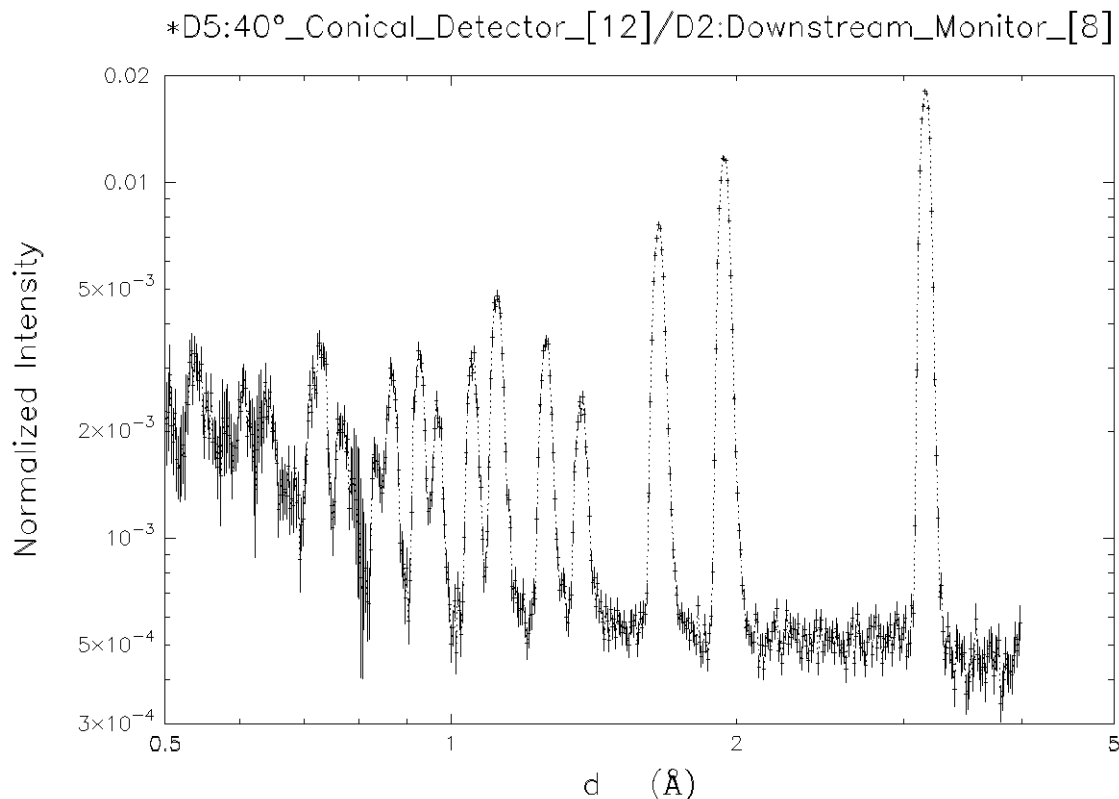


Figure 20. Reduced Si reflection data from the 40° detector bank (Fig. 19). Points are indicated by their vertical and horizontal error bars, and are connected by dotted lines. The largest d-spacing is Si(1 1 1), at 3.1355 Å. The automatically generated header indicates that the detector was divided by the downstream monitor (Fig. 18). Detector elements have also been divided by their solid angles, so the plot represents $d\Sigma/d\Omega$. Since the detector efficiencies are the same and the spectrum was measured simultaneously, no other normalizations were required.

operator. In particular, three essential pieces of information are *not* available within the .dat file written by MC_Run.

- 1) Is the detector Flat, or Curved? For the HIPPO model, choose Curved; that is, enter “N” when asked if the first is Flat. Thereafter the default will be Curved and you can answer blank.
- 2) What is the distance from the sample to the *center* of the detector? See last line of Table 11. The default will be the nearest distance to the first surface equation for the detector.
- 3) What is the Slope of the detector, dp/dZ ? See “Slope” in Table 11. Don’t forget the – sign.

All default parameter values are shown to you, and you may accept them by typing “/”. But you have to check the values – the defaults may not be right! To change a value that is not the first on a line, type “,” to skip preceding values and “/” to accept the remainder of the line.

Execute DLfromZt in a command-line window. Enter the name of the .dat file (including .dat). Skip the first two files, till you come to Transmitted_Spectrum_D2:Downstream Monitor. Accept the parameters for that file. Answering blank for the Output File Name is ok because the output data file will have “_DL.dat” appended to the input name. A .log file will also be written, showing all of your parameter choices (which can be compared to our choices in the HIPPO_data.zip file). Use blank again as the name of the next input file, because all the datasets are on the same file. As the possible ZT and RT files are presented, you can choose whether or not to convert each (default is “Yes”).

The resolution of the d-bins can never be finer than the t-bins, which is shown along with the binning defaults. We chose to keep the d-scale from 0.2–4.0 Å, with $d\lambda/\lambda$ exactly twice dt/t for each detector bank. For the 12° bank, enter “,, 0.016”. “/” the next line, “N” for the Flat question, and “/” the

following line. Then comes the question including sample-to-detector distance, L2. For the 12° bank, the answer is “, , 2.061”. “/” the next line, which brings up the Slope question; enter “, -0.577”. The calculation is then actually done! A final question is whether to accumulate this dataset with others. Your answer should be “N” because the detector banks in the HIPPO model have all different resolutions and d-bins.

Type blank for the next appropriate dataset, and repeat the list of questions. The two annular banks will be a little different. Instead of Flat, you will be asked if the detector is at Forward or Back angle. You must answer “B”. The Slope question is skipped, and L2 is correct. When done with all six banks, enter “/” for the file name to terminate the program.

You may plot the data with See_MC_Data. or you may export to other programs. The plot of the data from the 40° bank is shown in Fig. 20; the other plots are included in HIPPO_data.zip. Most of the plots have the vertical and horizontal error bars instead of symbols for the points, but the statistics are so poor on the two back-angle banks that “+” symbols are used and the error bars are omitted.

Here ends the tutorial. At this stage, you may wish to analyze the data in a program such as GSAS. The six data sets have been SWITCHED from Block-ASCII to column format and the files are included in HIPPO_data.zip.

XIII. In Case of Difficulty

<mailto:PASeeger@losalamos.com>

<mailto:lld@lanl.gov>

Appendix. Installation of NISP in a Local Computer

The user interface NISP_Win will execute *only* on a Windows operating system (Win98, NT, XP, or later). Full Fortran90 source code for the interface is provided, *but* it draws heavily on Windows APIs and the Microsoft Help library system. On the other hand, the runtime engine Mc_Run and the subroutine library MCLIB are highly transportable to any system with a Fortran90 compiler. You may thus either use a Windows platform to create a geometry file or write an *ad hoc* program to do it, and then run the actual simulation of the platform of your choice.

A. Download the Files

The full set of executable (Windows) and source files (Fortran90) may be downloaded from

<http://PASeeger.com>

The codes are updated frequently, and the dates of the latest revisions are shown on the home page; there is also a link to the complete revision history since December 1997. Clicking "NISPforWindows" will download a .zip file (currently 4.24 MB) that contains four executables, and all the files needed to compile if you wish to modify or add to the code. The source codes are also useful to examine the algorithms used in NISP, or to extract them for use in other codes. All source codes are fully commented to make them readable.

Documentation, including this manual, notes on code structures, conference proceedings, and published algorithms should also be perused and/or downloaded.

B. Extract the Programs from the .zip File

Use Win_Zip or equivalent. There are four executable files:



NISP_Win.exe to construct an instrument and generate a Geometry file



MC_Run.exe uses the Geometry file as input to run a Monte Carlo simulation



See_MC_Data.exe (OPTIONAL) plots the output data histograms of MC_Run



Super_Know.exe (OPTIONAL) makes histograms from a "Monitor" output file [note – the icon for Super_Know is a supernova]

The executables may be placed in any convenient directory. By default they will go to a folder \NISP\ within the folder where you extract. That is, if you extract to C: then a folder C:\NISP\ should be created.

Certain files and folders are **required** by NISP_Win. The program look for file Material.txt, first in the directory containing NISP_Win.exe, and then in C:\NISP\ or in a directory given in the environment variable SET NISP = ...

NOTE: the subdirectories **must** be in the same directory as file Material.txt. The data files are

```
x:\...\NISP\aaaread.me    (text file with installation instructions)
    \Material.txt        (REQUIRED)
    \Crystal.txt         (required for mosaic crystals)
    \Help\NISP.hlp      (REQUIRED)
    \NISP.cnt
    \Tables\*.tbl       (moderator energy/time tables)
    \Powders\*.pdr      (powder diffraction samples)
```

```

        \Crystals\*.lau (single crystal samples)
        \Kernels\*.sab (inelastic S(a,b) - these are long files you probably won't need)
C:\PGPLOT\grfont.dat (required to get labels on traces and graphs)
        \Get_PGPlot.txt
        \SYS_WIN\*.*

```

Programs MC_Run and See_MC_Data use PGPlot for graphics. To get any labels on the traces or graphs you have to have the font file that is appropriate to the Digital/Compaq/Hewlett-Packard Fortran compiler. (The proper version of the `grfont.dat` file is 66,020 bytes long.) The `\SYS_WIN\` folder has the required W9DRIVER for Windows. To recompile all of PGPlot, download the sources from the links given in `Get_PGPlot.txt`.

C. Source Files

The remaining subdirectories in the `.zip` file contain all the sources and the resources necessary for compilation. If you have the space available (approximately 4.4 MB), I recommend you go ahead and extract them. They are useful in several situations:

- For documentation of the algorithms – never trust a black box.
- For debugging – I really appreciate it when you can tell me *exactly* where I messed up!
- To use “Super_Know” to get information not available in ordinary detectors, you will have to modify the sample code provided to get exactly what you want.
- If you do not have PGPlot, and do not wish to install it, you may wish to recompile MC_Run with all references to PGPlot removed. There are comments within the code indicating what to delete.
- So that you can add new features to the code – then please let me know so I can make the new features public.

D. Compiling with Visual Fortran

The Fortran compiler in the Microsoft Visual Developer Studio environment has changed ownership several times: Microsoft, Digital, Compaq, and Hewlett-Packard. We show the Digital version 6.0, and you should note that in other versions the steps will probably not be exactly as shown here. Copy the “include” files to `C:\Program Files\Microsoft Visual Studio\DF98\Include` (or equivalent folder).

- a) Create a Workspace with the name NISP
File > New > Workspace tab, Workspace name = ‘NISP’, OK
- b) Create a Project called MCLIB
File > New > Projects tab > Fortran Static Library, Project name = ‘MCLIB’, Add to current workspace, OK, OK
- c) Insert all of the library modules into the project
Project > Add to project > Files, Look in: `C:\NISPMCLIB\`, select all *.f files, OK
- d) Select and Compile the “Release” version of the library
Build > Set Active Configuration > MCLIB – Win32 Release, OK
Build > Build MCLIB.lib
(If you ever need to run the Debugger, you will need to repeat this step for the “Debug” version.)
- e) Create another project called MC_Run
File > New > Projects tab > Fortran Console Application, name = ‘MC_Run,’ Add to current workspace, An empty project, Finish, OK
- f) Insert the source code into the project (*Note: select .for, not .f*)
Project > Add to Project > Files, Look in: `C:\NISP\`, select MC_Run.for, OK
- g) Specify that this project “depends” on the library

Project > Dependencies > MCLIB, OK

- h) Set compiler options. For version 5.0, Digital changed the default value for the units of the RECL parameter from bytes to 4-byte words. The easiest way to account for this is to be sure RECL=13 in all OPEN statements for .mon files.
- i) Compile and Link the “Release” version
Build > Set Active Configuration > MC_Run – Win32 Release, OK
Build > Build MC_Run.exe
- j) *If you want to compile a “Debug” version of MC_Run,*
Project > Settings > Fortran tab > Settings For: Win32 Debug, Category: Debug, Compile D Lines, OK
Build > Set Active Configuration > MC_Run – Win32 Debug, OK
Build > Build MC_Run.exe

E. Analysis Programs

Fortran source codes and Windows executables are available for some sample analysis programs. From <http://PASeeger/downloads>, download the file Analysis.zip. The contents of the file may vary from time to time. The first two programs you will need are

- SWITCH Convert 1-D data (Block-ASCII format) to columns. The header is retained. Bin boundaries are replaced by “POINTS” at the geometric mean of the bin.
- SPREAD Convert 2-D data (Block-ASCII format) to spreadsheet layout. The header is retained. One array gives histogram values (Y\X DATA) and a second gives standard deviations (Y\X STDDEV). The arrays have the means of the X bins on the first row, and the means of the Y bins in the first column.

These two programs may be used to convert the data, and may also be used as examples of how to read the Block-ASCII data format.

The following data reduction programs convert from detector position and time of flight to physically meaningful variables, such as wavelength, momentum transfer, or energy transfer. They include all recognized normalizations and corrections, and implement full propagation of errors from the statistics of the detector histograms and from any rebinning operations. Operations are recorded algebraically in the header (requiring external subroutines NEWHEAD, OPLIST, and PREMUL, included in the .zip file). Other external subroutines come from MCLIB, or are also in the file.

- QLfromXYt Convert from 3-D XYt data to 2-D Q - λ histograms, and summed over λ for 1-D Q histograms. Derived from a small-angle scattering code, but good at all angles. May be used as example of reading 3-D data.
- QLfromRt Read histograms in Radius and time, and rebin in Q - λ . May generate a “figure of merit” based on omitting data from bins with inadequate resolution at smallest angles.
- DLfromZt Reduces diffraction data from flat or curved detectors recorded in the Z-direction or from annular detectors (at either forward or back angles) recorded in the R-direction. Converts position/time to d/λ and sums over λ .
- EEfromRt Convert (R, t) bins to (E_2 , E_1) bins, rebin vs. ΔE , and accumulate. This version tailored specifically to VISION (or TOSCA). Data and variances from multiple detector banks may be summed.

F. Data Files for the HIPPO Tutorial

If you run the HIPPO tutorial (Sec. XII) as a test case, you can compare your results to files extracted from [Downloads](#): HIPPO_data.zip.

G. Executing MC_Run in Batch Mode

Two files must be constructed in order to execute multiple simulations in batch mode: an input file with the run-time parameters for all runs, and a .bat file that executes MC_Run one or more times. A .bat file can be executed from an MS-DOS command-line window by typing its [path\]name. To run it in background while performing other tasks, enter the command

```
START/LOW ...bat
```

Here is an sample .bat file:

```
cd \VISION
set FOR_READ=TestFour.in
\nisp\MC_Run
set FOR_READ=
```

The first command changes the directory to the location where all of the files are found for the problem being simulated (this saves having to include the path on all references to the files). The second command redirects the Fortran standard input device, so that all requests from Fortran for input from the console will be read from your file instead. The next command executes MC_Run. Finally, the Fortran standard input device is restored to its default.

The input file, in this case `TestFour.in`, may include any number of geometry files and the appropriate parameters from the namelist shown in Sec. XI.A (Table 4). An additional “feature” that you have to watch out for is that you may be asked whether a detector bank is to be recorded in 3-D. In this example an extra line is included with “N” as the answer in case the question is asked; the presence of any number of lines in the file before the “\$RUN” will be ignored if not needed. Suppose that four cases of a set of interpolated variables are to be simulated. The input file might look like this:

```
VISION_mosaic1
Mosaic015
N          /3D histogram
$RUN  nsplit=10, iseed=0 /
10000
VISION_mosaic2
Mosaic030
N          / no 3D (if asked)
$run/
15000
VISION_mosaic3
Mosaic045
N          /3D output
$RUN/
20000
VISION_mosaic4
Mosaic060
N          /3D output
$RUN/
30000
```

The first line for each case is the geometry file, and the second line is either blank (for same name as the geometry file) or a name for the output files. The namelist for the first job sets `NSPLIT=10` and `ISEED=0` for randomization; for subsequent jobs the default parameters will be the same as the previous job, and the random-number sequence will continue, so no parameters have to be entered. Different numbers of histories to be detected were chosen for the four cases because the detection rate depends strongly on the parameters being varied.