

The Neutron Instrument Simulation Package NISP: Recent Developments

Philip A. Seeger^{1,2}, Luke L. Daemen²

¹*Consultant, 239 Loma del Escolar, Los Alamos, NM 87544, U.S.A.*

²*Manuel Lujan Jr. Neutron Scattering Center, Los Alamos National Laboratory,
Los Alamos, NM 87545-1663, U.S.A.*

Abstract

A review is given of the developments in the Los Alamos Neutron Instrument Simulation Package (NISP) since the ICANS-XIV meeting (1998). New element types include a Fermi chopper, materials with nuclear resonances (spin dependent), and segmented guides. The mosaic crystal algorithm has been improved, and more consideration has been given to multiple scattering. Magnetic induction field regions that overlap material regions have been added, along with a variety of field definitions such as uniform, gradient, current loop, and solenoid. Spin precession is computed when a neutron moves in a field region, and several region types now include polarization. An example of a benchmark simulation of the H8 triple-axis spectrometer is shown; this model has been used to intercompare various codes. Although the main user interface of NISP continues to be the web application, <http://strider.lansce.lanl.gov/NISP/Welcome.html>, a stand-alone interface for Windows (98 and NT flavors) has been posted at the web site. This NISP_Win interface does *not* include the database and visualization features of MC_Web, but it is easier to modify and update, or to add individual features. We continue to support users, and would be most happy to assist in adding new algorithms to NISP. All source codes are fully available.

1. 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. As is typical of Monte Carlo codes, precision may be enhanced by running many histories, but accuracy is always limited by the algorithms used. It is our intention that the code will be completely versatile and that it will continue to grow as users contribute new and improved modules.

NISP has been described in detail in previous ICANS Proceedings [1–3]. We will give only a brief description here, and then discuss the new features. The package has three components:

- The user interface, MC_Web (or the new NISP_Win described in sec. 5), used to build an instrument by selecting available beamline elements and providing relevant parameters. Generates a geometry file with all parameters in structures.
- The Monte Carlo engine, MC_Run, uses the geometry file for input, allows choice of run-time parameters, and generates detector histograms and statistical summary data.

E-mail: PASeeger@aol.com. Please include “NISP” in the subject.

Keywords: Monte Carlo, simulation, polarization, solenoid

- The subroutine library, **MCLIB**, includes algorithms for neutron transport in many types of material and magnetic induction regions, and associated utilities.

MCLIB

The present **MCLIB** is descended from a library of Fortran subroutines written at the Rutherford-Appleton Laboratory [4]. Significant additions and revisions were made and the entire code was rewritten in a structured form in 1994 [1]. The library currently contains more than 80 procedures, including vector analysis and neutron transport, material properties and scattering functions, time-dependent devices, magnetic induction fields such as current loops and solenoids, spin transport, random distributions, and general utilities. Subroutine `OPERATE` and its other entry point `EXIT_REG` play special roles in the transport process [3]. This procedure is called to determine what happens whenever a neutron enters a new region. Possible results from `OPERATE` include: detection or loss of the neutron, exiting the region with reduced statistical weight (partial absorption), splitting the neutron into two histories with the sum of statistical weights equal to the original weight, or a transform of the coordinate system. Coordinate transforms are applied for region types made up of sub-regions, elements that change the beam direction, and time-dependent devices. Such coordinate transforms may be “undone” when the region is exited.

MC_Run

When the code was restructured, the application `MC_Run` was extracted from the extant single-purpose codes as a general program using a geometry file for its input [1,3]. `MC_Run` transports neutrons in the instrument, averages over histories, and produces a variety of output files with histograms and information about the course of the simulation. When executed, `MC_Run` first asks the user for the geometry file name. After the geometry file is read, `MC_Run` prompts the user for parameters such as the type of intermediate information to be saved, the variance reduction techniques to be applied, random number seeds, *etc.*, using Namelist I/O as in the following box. To modify any parameters, the user types “ \$RUN ” followed by “keyword=value, keyword=value, . . .” and ending with “ \$”. New parameters are number of times to `REUSE` an input file, and the option to randomize `ISEED`. If nothing needs to be changed, enter “ \$RUN \$”. Then enter the number of histories to be detected.

```

$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
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 induction fields ('Y' for yes)
REUSE = 1         Allowed number of times to reuse input source file
ISEED = 12345678  Random-number seed, 0 to randomize
$

```

MC_Web

The initial user interface [2] has become a full web application (`MC_Web`) including database management and instrument visualization. Significant features from the computer science point of view [5] are:

- Combination of very different technologies: Fortran vs. Web/Smalltalk
- Robustness; has functioned without significant downtime for the past five years with minimal support or maintenance.
- Meta programming.
- 3D visualization using VRML.
- Centralizing scientific element type definitions.

Users enter the application at <http://strider.lansce.lanl.gov/NISP/Welcome.html>. There is an extensive list of element types, such as moderators, samples, filters, choppers, guides, etc. During the creation of an instance of an element, the user is presented with one or more forms such as the example shown in Fig. 1 for a Mosaic Crystal. The parameters are divided into extrinsics (that specify the position and orientation in space of the component) and intrinsics (that specify the shape, size, physical characteristics, and parameters needed by the particular algorithm.) Input may be numerical data, as well as options selected by buttons or pull-down menus. Help is available at each entry box on the form to provide definitions for the

Element Type: <input type="text" value="Mosaic Crystal (Flat)"/>
Element Name: <input type="text" value="Crystal (left)"/>

To edit the parameters value, change the data below and press Update at the end of this form.

Extrinsic Parameters:

x: <input type="text" value="-274.53"/>	mm
y: <input type="text" value="0"/>	mm
z: <input type="text" value="-0.256"/>	m
Horizontal Tilt angle: <input type="text" value="0"/>	°
Vertical Slope angle: <input type="text" value="0"/>	°

Intrinsic Parameters:

Thickness: <input type="text" value="3.0"/>	mm
Width: <input type="text" value="100.0"/>	mm
Height: <input type="text" value="100.0"/>	mm
Axis orientation: <input checked="" type="radio"/> Flat <input type="radio"/> Horizontal <input type="radio"/> Vertical	
Radius of curvature: <input type="text" value="0"/>	m
Crystal parameter file: <input type="text" value="PG_002"/>	
Spread of lattice plane spacings: <input type="text" value="1.0e-4"/>	
Horizontal component of mosaic spread: <input type="text" value="0.022"/>	
Vertical component of mosaic spread: <input type="text" value="0.022"/>	
Crystal cut angle, chi: <input type="text" value="0"/>	°
Crystal plane rotation, psi: <input type="text" value="0"/>	°
Crystal Temperature: <input type="text" value="300.0"/>	K
Rotate Z-axis after reflection?: <input type="radio"/> Yes <input checked="" type="radio"/> No	

Figure 1: Typical MC_Web form, to define a Mosaic Crystal.

parameters and to assist the user in selecting proper values for these parameters.

Additional details of the program features can be found in ref. [6]. A complete “How to . . .” instruction manual is available at the web site [7]; all users are urged to read this document. All questions and requests for help should be sent by e-mail to *PASeeger@aol.com* and/or *lld@lanl.gov*, and should include “NISP” in the subject line. We will be particularly pleased to assist in adding new algorithms and elements to NISP.

2. New and revised elements

Modifications and additions to MCLIB and to MC_Run are made fairly often. These are reported in the “What’s New” button on the web site, and the revised codes are included in the “beta_version” folder. Approximately yearly the release version of the package is updated to include new features up to that date, and the immediate preceding version is preserved in its own folder. Unfortunately we do not have the support to make changes in MC_Web. We are grateful to Thierry Thelliez [5] for continuing to make changes we consider essential. Many of the new features can only be incorporated in simulations by using the “General Region” type and entering parameters directly, or by using the NISP_Win interface described in sec. 5 below. All source codes are available at the web site [8].

Nuclear resonances

The table of predefined materials now includes materials with nuclear resonances, such as cadmium, gadolinium, samarium, gold, ^{238}U , and ^{240}Pu . The nuclear resonance materials are listed (along with others) when the pull-down menu of materials is selected from many element forms. The algorithm is a generalized one-channel R-matrix (isolated *s*-wave resonances) [9]. The model reduces to the more familiar Breit-Wigner form, but also allows non-resonant contributions and interference terms, and includes multiple isotope-spin channels. For instance, elemental Gd has 2 resonances, and Sm has contributions from non-resonant isotopes in addition to the strong resonance in ^{149}Sm at 97.3 meV. Cadmium is included because pulsed sources generally include neutrons with energies well above its 178 meV resonance; if the source is limited to wavelengths longer than about 0.5 Å, then you may usually treat Cd as a perfect absorber. The subroutine `SINGLELEV` computes the various cross sections, and the complex scattering length.

If a material with non-zero spin (such as Sm or ^{149}Sm) is placed in a magnetic field it is assumed to be fully polarized, and the cross sections and scattering lengths for spin-up and spin-down neutrons are calculated separately. Thus neutrons may be polarized by transmission or by reflection.

Mosaic crystals

Our algorithm transports neutrons stepwise through a crystal, with an exact analytic solution of the Darwin equations at each step [10,11]. At each step the neutron may be reflected, transmitted, or scattered incoherently, and absorption is included by reducing the statistical weight. The procedure is a logical function, `LMONOCRM`, which is set to false if the number of reflections is 0 or even, or if the neutron scattered incoherently. This allows a neutron to be flagged as “bad.”

There are several advantages to taking small steps instead of using the analytical solution for the total crystal thickness:

- The depth of the interaction(s) within the crystal is estimated
- “Walking” along the crystal due to multiple reflections is included
- Secondary extinction is included naturally
- The shape of the crystal is not limited to plane parallel surfaces.

All geometric effects, including mosaic spread, are treated externally as part of the neutron transport. Both reflection (Bragg) and transmission (Laue) geometries are supported. The parameters are shown on the sample form in Fig. 1. Special features include flat or cylindrically bent crystals (there is a separate element type for spherical surfaces), plane orientation that does not have to be parallel to the surface, and temperature dependence. There is a file with parameters for a variety of crystals and planes; in the example PG002 has been selected. The last entry on the form allows the direction of the beamline axis to be rotated after reflection by a monochromator.

Fermi chopper

A Fermi Chopper has a body that is a vertical cylinder with a central opening containing a set of vertical curved (or straight) collimating blades. The combination of high rotation speed and blade curvature acts both as a pulse generator and a narrow velocity selector. For a neutron sufficiently fast that the path through the chopper can be approximated as circular, the relationship between radius of curvature, frequency, and the nominal neutron velocity is $v_0 = 4\pi Rf$. Figure 2 shows the dialog box for the Fermi Chopper in the NISP_Win interface. By default the rotor is assumed transparent and the blades are perfect absorbers, but in this example the rotor material has been set to “Attenuator” and the macroscopic cross section terms of Al have been entered, and the blades are the “Resonant” material Cadmium, as described above. (At the time of writing, the nuclear resonance option was not provided in the MC_Web interface.) Note that the extrinsic X_0 is offset by the sagitta.

The procedure `FERMI` is a logical function, set to true if the neutron passes through a single slit in the proper direction (if the blades are straight, then either the nominal phase or a 180°

Fermi Chopper	
Element Name	Fermi Chopper
Extrinsic Parameters	
Center of Rotor, X (mm)	0.5
Y (mm)	0.
Z (m)	5.5
Intrinsic Parameters	
Rotation Frequency (Hz)	300.
Curvature (reciprocal radius) of blades (1/m)	0.862
Nominal time-of-flight from source (μs)	1275.
Phase jitter (μs)	1.5
Rotor: Diameter (mm)	110.
Height (mm)	150.
Rotor material is:	<input type="radio"/> Transparent <input type="radio"/> Absorber <input checked="" type="radio"/> Attenuator
High-energy macroscopic cross section (1/cm)	0.085
2200-m/s macroscopic cross section (1/cm)	0.187
Slit Package: Width (mm)	20.
Height (mm)	50.
Length (mm)	100.
Blades: Number	10
Intrinsic Parameters (continued)	
Thickness (mm)	0.1
Blade material is:	<input type="radio"/> Absorber <input type="radio"/> Attenuator <input checked="" type="radio"/> Resonant
High-energy macroscopic cross section (1/cm)	0.
2200-m/s macroscopic cross section (1/cm)	0.
Material (Resonant Only)	Cadmium
OK Cancel	

Figure 2. Entry form for Fermi Chopper parameters, using the NISP_Win interface.

rotation is considered good), or if it is absorbed. We first determine if the neutron is ever vertically within the limits of the slit package, and if so find the chopper phase angle ϕ (including a Gaussian jitter term) for the midpoint of the trajectory, defined to be 0 at the nominal time of flight. We then consider the two surfaces representing the *sides* of the slit package. We express the distance D of a neutron at $(x = X - X_0, z = Z - Z_0)$ with velocity (v_x, v_z) from a rotating vertical cylindrical surface (and its time derivatives) as

$$D = \frac{K}{2}(G^2 - x^2 - z^2) + x \cos \phi + z \sin \phi - G, \quad (1)$$

$$\frac{dD}{dt} = (\cos \phi - K x)v_x + (\sin \phi - K z)v_z + (z \cos \phi - x \sin \phi) \Omega, \quad (2)$$

$$\frac{d^2D}{dt^2} = -\frac{K}{2}(v_x^2 + v_z^2) - [v_x \sin \phi - v_z \cos \phi - (x \cos \phi + z \sin \phi) \Omega] \Omega, \quad (3)$$

where the parameters of the surface are curvature K (m^{-1}), distance from origin G (m), and rotation speed Ω (rad/s). The final term in eq.(3) is a Coriolis term. For the sides, $K = 0$ and $G = \pm W/2$. If the neutron trajectory was not completely within the slit package (either horizontally or vertically) we find the times of entry and exit. We write the trajectory as the distance from the center of the outermost blade, normalized to the separation of blades so that a near-integer value represents being inside a blade. If the min and max are not in blades and the integer parts are the same, then the neutron passes unattenuated. Otherwise if the blades are perfect absorbers the weight is set to zero, or else the distances traveled through the body and through blades are found and the appropriate attenuation factors are applied. The material between the blades is transparent, and scattering is not currently included in the model.

Segmented guides

In addition to element type Guide (Straight) there is now a type Guide (Segmented/Curved). The curved guide must be horizontal and aligned with the Z-axis, and cannot converge or diverge. Parameters include the total bend angle and the number of segments, which are identical straight segments; only the first segment is actually defined in the geometry file. By default the material in the interior of the guide is Void, but it may be changed to 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!).

As a neutron passes each segment, the coordinate system is rotated and it reenters the same region definition. If the neutron reaches the exit of the guide, the Z-axis will have been rotated by the total bend angle. Neutrons that penetrate a 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 (unless the substrates are perfect absorbers). The algorithm for guides is coded directly in the OPERATE subroutine.

Multiple scattering

It is very common to use Monte Carlo integration to estimate multiple scattering corrections in neutron experiments. NISP includes various sample types and materials, many of which also account for multiple scattering. By combining the simulation of beam optics and multiple scattering, significant correlations are included [12].

There are two ways to apply probabilities when a random choice is to be made between possible processes. One is “analog” Monte Carlo, in which the outcome for each neutron is selected by comparing a uniform random deviate to the relative cross sections for the various processes. In NISP this method is used for the class of simple materials that are used for structural members and filters. Since the principal use of such materials is absorption, we only use computer time to track a small fraction of non-absorbed histories. The other method, “weighted” Monte Carlo, is to split the incident neutron history into two or more fractional parts with appropriate statistical weights, and to track all possibilities. This is done in NISP for the class of samples. Thus we don’t lose histories due to absorption in the sample or by transmission; *every* neutron that has managed to reach the sample through the incident collimation system is forced to scatter at least once, and continues with its weight reduced to account for transmission and absorption.

The algorithms in NISP for simulating scattering in various materials or samples have been given in detail in ref. [12] and need not be repeated here. Most are simple and fast, designed to measure instrument resolution and to approximate backgrounds in a variety of configurations. For instance, simple materials derive the scattering from the complex scattering-length density, and nuclear-resonance materials find the scattering length from resonance parameters. Scattering samples include small-angle, inelastic isotropic, an $S(\alpha, \beta)$ model, and powder diffraction.

3. Magnetic induction, and precession

The increasing importance of polarization in neutron scattering instrumentation for condensed matter research means that Monte Carlo design tools must be able to track neutron spin during neutron transport. In particular, we must be able to solve Bloch’s precession equation for arbitrary magnetic induction configurations, including time-dependence. Since simulations require averaging a large number of neutron histories, the computational procedure must be fast, as well as accurate and precise. Such an algorithm [13] has been incorporated in NISP.

Neutron beam polarization is represented by a 3-vector \mathbf{P} , with magnitude $P \leq 1$ being the degree of polarization [13,14,3]. The probability that a particular neutron in the beam has its spin in the direction of \mathbf{P} is $(1+P)/2$, and the probability of the opposite spin is $(1-P)/2$. Thus $P = 0$ is an unpolarized beam, with probability in *any* direction being 50%. Because of the quantum-mechanical nature of spin, any beamline component that is sensitive to neutron spin will have a known orientation, described by a unit vector \mathbf{n} . The probability that a neutron interacts with spin parallel to \mathbf{n} is $(1+\mathbf{n} \cdot \mathbf{P})/2$ and the probability for anti-parallel is $(1-\mathbf{n} \cdot \mathbf{P})/2$. In a Monte Carlo simulation, the component may *either* split the neutron into two independent histories with $\mathbf{P} = \pm \mathbf{n}$ and statistical weights multiplied by the respective probabilities, *or* it may use a random number to select one spin or the other based on the relative probabilities. Any operation that is symmetric with respect to spin, such as precession or spin-flip, can be applied either to the beam polarization or to the separated histories.

When transporting neutrons with non-zero values of \mathbf{P} , NISP checks for the presence of a magnetic induction (\mathbf{B}) field. If there is a field, then the line integral of $B(t)$ along the trajectory is computed:

$$S(t) = -\gamma_n \int_0^t B(\tau) d\tau \quad , \quad (4)$$

where $\gamma_n = -183.247 \text{ rad} \cdot \mu\text{s}^{-1} \text{ T}^{-1}$ is the neutron gyromagnetic ratio. Using S as the independent variable, and decomposing $\mathbf{P}(S)$ into components parallel and orthogonal to

$\mathbf{B}(S)$, the rapid precession is separated from the slowly varying direction of \mathbf{P} . That is, the “stiffness” of the Bloch equation is removed, and numerical integration is practical [13]. For further details, see the source codes for procedures `BFIELD`, `BINTEGRAL`, `RK4BLOCH`, and `RKPRECES` [8].

Magnetic field regions in NISP coexist with and overlap material regions, and they may or may not have common surfaces. They do not have to fill all space, and they are never “connected” to each other or to material regions except for sharing surfaces. In the region matrix they have surface designators +10 or -10. Several field algorithms (class 60) have already been included or are planned in MCLIB:

- Uniform (60)
- Linear (61.0) or Helical (61.1) gradient between two (quadratic) surfaces
- Multiple circular current loops (62), including point dipoles as a limit
- Solenoid (63)
- Set of parallel finite current sheets (64); not yet implemented
- Field defined in a file (65); file format not yet defined
- Vector sum of any number of other types (69)

Each algorithm is in a function procedure attached to the `BFIELD` source file. Additional subroutines called are `BLOOP` and `SOLENOID`. In each case a pulsed-source time dependence may be specified; that is, the field may be made proportional to the nominal neutron velocity over a specified time range.

At the present time the magnetic regions have not been explicitly implemented in MC_Web (but *are* in NISP_Win, see below). To create a field region in MC_Web, choose a pipe, cylinder, or block to define the geometry, and then choose “General Region” for the material type. Write a text file with your parameters, and use cut-and-paste to enter it in the form. For a listing of all parameters used for each type, see the file `MC_ELMNT.INC`, or the excerpt of that file in Appendix A. The geometry file will have to be edited. Find the region number by counting down in the list of region names, and find the corresponding row in the region matrix. Change 1 to 10 and -1 to -10 wherever they occur in that row.

We wish particularly to mention the solenoid routine. To the best of our knowledge this is the first time a closed analytic form for the field surrounding a solenoid has been given. We were unable to find such a form in the literature; Durand [16] gives the vector potential and shows the derivative necessary to get the radial component, but not the axial component. The result involves complete elliptic integrals of the 1st, 2nd, and 3rd kinds, denoted respectively $K(k)$, $E(k)$, $\Pi(n, k)$. The modulus k and the characteristic parameter n depend on the geometry of the problem. Consider a thin solenoid of radius a (m) and length $2b$ (m), with current density J (A/m). In a cylindrical coordinate system centered on the solenoid, let (ρ, z) be the point at which components B_ρ, B_z of the vector \mathbf{B} (T) are to be calculated. Using subscripts 1 and 2 to represent the entrance and exit surfaces of the solenoid, find the distances to the far “corners,” and the respective moduli of the elliptic integrals:

$$R_1 = \left[(a + \rho)^2 + (z + b)^2 \right]^{1/2}, \quad k_1 = 2\sqrt{a\rho}/R_1, \quad (5)$$

$$R_2 = \left[(a + \rho)^2 + (z - b)^2 \right]^{1/2}, \quad k_2 = 2\sqrt{a\rho}/R_2. \quad (6)$$

We find from ref. [16] that the radial component depends on elliptic integrals of the 1st and 2nd kinds, and is a difference of terms from the two ends of the solenoid:

$$B_\rho = \frac{\mu_0 J}{2\pi} \left\{ \frac{R_2}{\rho} \left[\left(1 - \frac{k_2^2}{2} \right) K(k_2) - E(k_2) \right] - \frac{R_1}{\rho} \left[\left(1 - \frac{k_1^2}{2} \right) K(k_1) - E(k_1) \right] \right\}, \quad (7)$$

where $\mu_0 = 4\pi \times 10^{-7}$ T-m/A is the permeability of free space. The axial component will need integrals of the 3rd kind, for which the characteristic parameter is

$$n = -4a\rho / (a + \rho)^2, \quad (8)$$

which is independent of z and hence the same for either end of the solenoid. Using properties of the elliptic integrals from ref. [17], we arrive at a similar form for the axial component, involving elliptic integrals of the 1st and 3rd kinds:

$$B_z = \frac{\mu_0 J}{2\pi} \left\{ \left[C_2 \mathbf{K}(k_2) - \frac{z-b}{R_2} \left(\frac{a-\rho}{a+\rho} \right) \mathbf{\Pi}(n, k_2) \right] - \left[C_1 \mathbf{K}(k_1) - \frac{z+b}{R_1} \left(\frac{a-\rho}{a+\rho} \right) \mathbf{\Pi}(n, k_1) \right] \right\}. \quad (9)$$

The factors C_1 and C_2 are quite complex, involving (among other terms) partial derivatives of k and ϕ (a phase angle related to n) with respect to ρ (at constant z).

$$\frac{\partial k_1}{\partial \rho} = k_1 \left(\frac{1}{2\rho} - \frac{a+\rho}{R_1^2} \right), \quad (10)$$

$$\frac{\partial \phi_1}{\partial \rho} = \frac{1}{k_1(z+b)} \left(\frac{a^2 - \rho^2}{R_1^2} + \frac{(a-\rho)^2}{(a-\rho)^2 + (z+b)^2} - \frac{2a}{a+\rho} \right), \quad (11)$$

$$C_1 = \frac{z+b}{R_1} \left[\frac{a-\rho}{a+\rho} - \frac{a+\rho}{2\rho} + \frac{(a-\rho)^2}{2\rho k_1(1-k_1)^2} \frac{\partial k_1}{\partial \rho} \right] + \frac{(a-\rho)^3}{4\rho\sqrt{a\rho}} \frac{\partial \phi_1}{\partial \rho}. \quad (12)$$

The expression for C_2 is similar, with $(z-b)$ in place of $(z+b)$. There are many discontinuities and special cases for these expressions, which can be found in the `SOLENOID` source code [8]. The code to compute the elliptic integrals may be found in `CMPLT_PI`.

Figure 3 illustrates the field map for a solenoid. The solid lines are contours of the vector potential (subroutine `ASOLND`). It can be seen that the vector plot agrees with the field lines, assuring us that we have correctly taken the derivatives. We have also tested by comparing to an approximate solenoid formed by a large number of current loops.

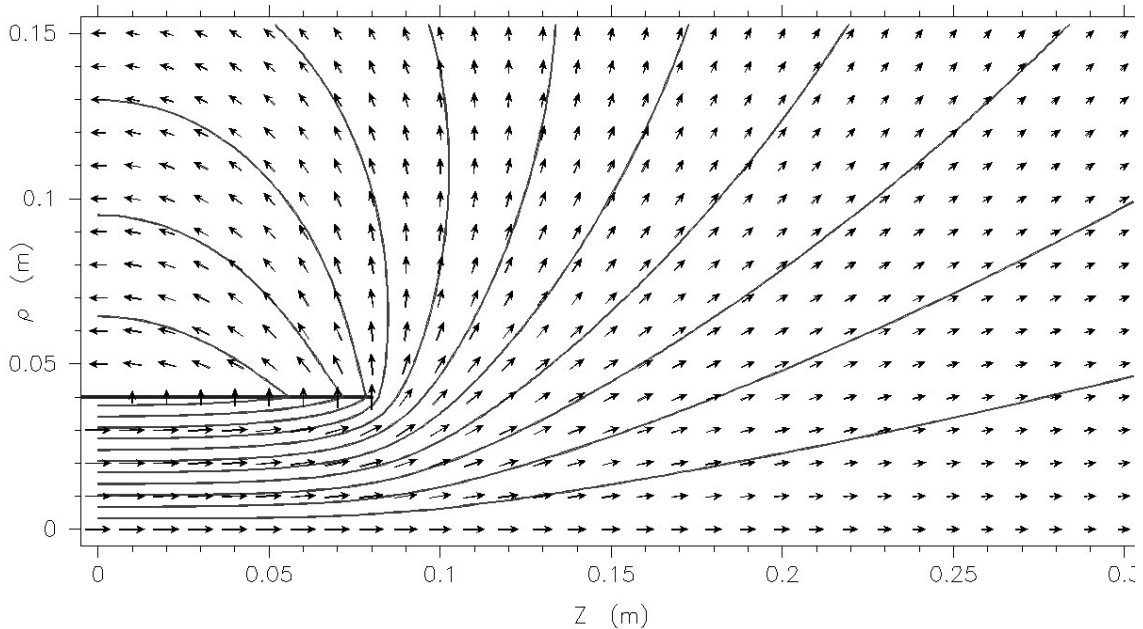


Figure 3: Analytic calculation of induction field of a solenoid (vectors), compared to flux lines plotted as contours of the vector potential. The lengths of the plotted vectors are logarithmic: compared to the vector at the origin, $3/4$ length is 10%, $1/2$ length is 1%, $1/4$ length is 0.1%, and 0 length is $<0.01\%$.

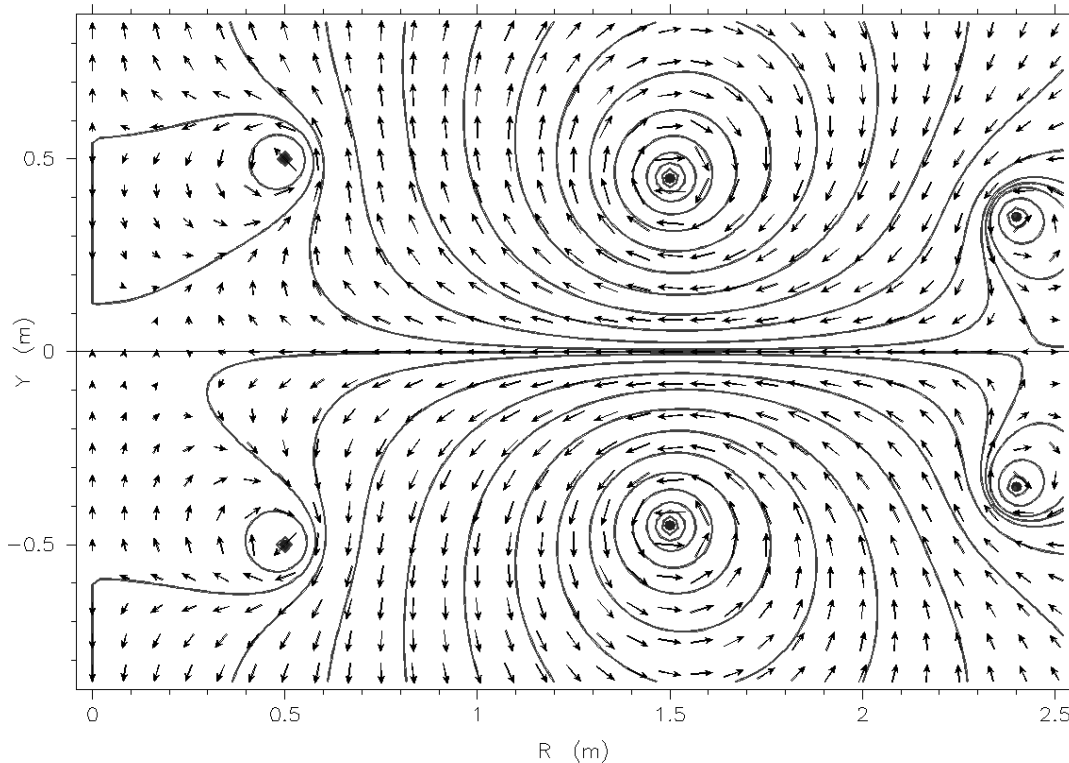


Figure 4: Magnetic induction fields in the SPAN instrument. Horizontal coils are arranged in 3 pairs. For the main coil in the center and the compensation coil at the right, currents in the coils above and below the midplane are equal and opposite. The secondary coils at the left have both symmetric and antisymmetric current components, to shape the field at the origin (the sample position).

Another example of a field calculation, with 6 horizontal current loops, is shown in Fig. 4. This represents the SPAN instrument at HMI [18]. As in Fig. 3, the lines are contours of the vector potential (subroutine `ALOOP`). It would take a very long execution time to do a full Monte Carlo simulation of precession through this field, but by making a number of runs with specific source points and orientations it is possible to study effects due to sample size and beam divergence.

4. Benchmark simulation of H8 TAS

A recent study [19] compared five Monte-Carlo codes simulating the same instrument, the (former) H8 triple-axis spectrometer at the Brookhaven reactor. The result was that if we carefully match algorithms, we can get the same result (at least to first order), as seen in Fig. 5. For instance, in NISP the blades of the Soller collimators were made perfect absorbers instead of a real material. Perhaps the greatest triumph was matching the variety of mosaic crystal algorithms. This was done by having each author set parameters to reproduce the peak reflectivity and width of the rocking curve for the PG002 crystals [20]. This emphasizes that the parameters we call “mosaic spread” are *not* the same as a measured rocking curve. The conclusion of the study was a general satisfaction with the level of agreement, and the statement that reasonable statistics can be obtained with any current package on today’s computers in times less than an hour. Thus parametric studies to optimize instrument designs are practical.

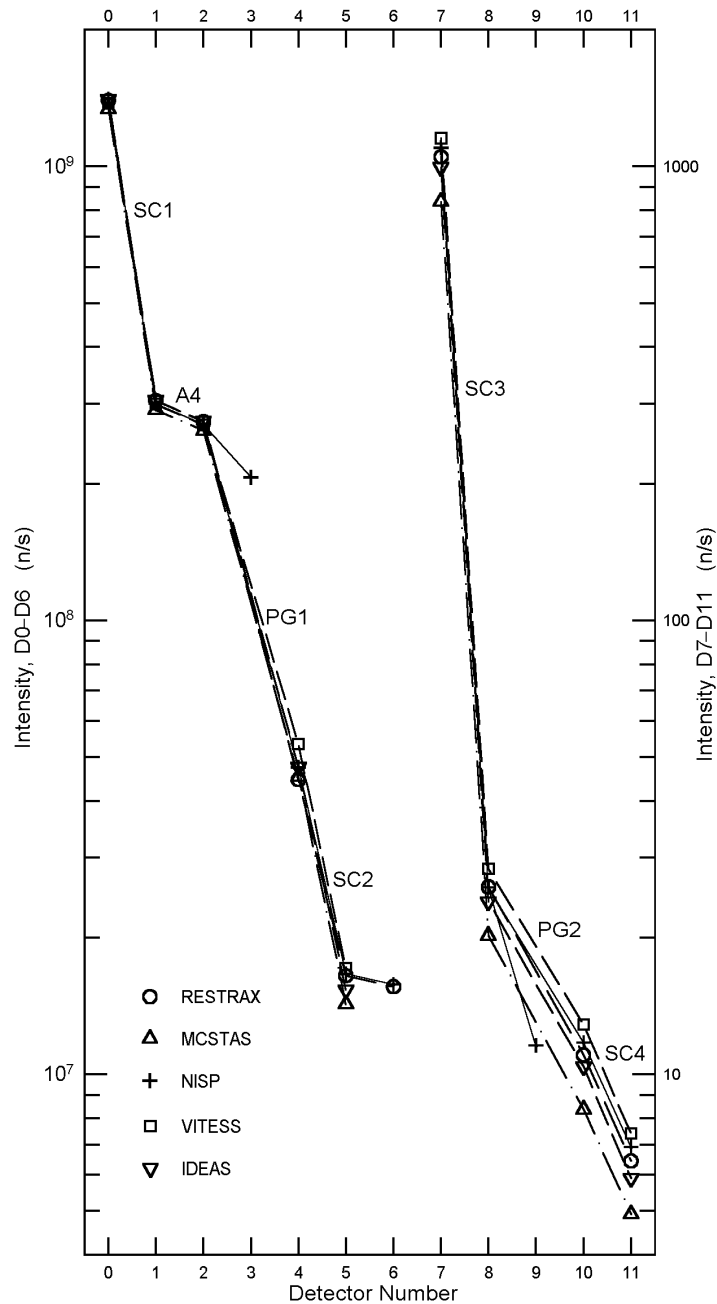


Figure 5: Results of benchmark simulation of H8, intensities at surfaces between beamline elements. SC1–SC4 are Soller collimators, A4 is an aperture, and PG1–PG2 are mosaic crystals. The vanadium sample is at position D5; D6 is the transmitted and D7 the scattered beam (not all codes report D6, and only NISP shows the transmitted beams through the crystals). The largest difference between codes is the V scattering probability. See ref. [19] for descriptions of the various codes.

A report with a full set of acceptance diagrams and profiles for this simulation, along with examples of other information that can be obtained only by Monte Carlo, is available at the NISP web site [21]. Acceptance diagrams produced by Monte Carlo show variations in phase-space density (such as fuzzy edges). A feature found only in NISP is backscatter from the crystals, which can be seen in detectors preceding the crystals. This is because only NISP treats the connections between elements as a matrix instead of a linear array.

5. Windows interface, NISP_Win

Because web access can be very slow, especially from Europe, users have requested a stand-alone interface independent of the LANSCE web server. Toward this goal we have adapted the Fortran element-definition routines to generate dialog boxes (*e.g.*, Fig. 2) and have combined them in a QuickWin Fortran application, NISP_Win. This runs under Windows 98 or later, and NT 4 or later; it is *very* dependent on Windows features. A complete package of executables, data files, help files, and all sources can be downloaded from the NISP site [22]. Installation instructions are in the file `aaaread.me`. The visualization and database management features of MC_Web have *not* been emulated. New or different features include:

- Instrument definition file, *.inst, includes all parameters in Namelist format, and the connection matrix (optional).
- Either a .inst file or a .geo file (if created by NISP_Win) may be used as input.
- Single connection matrix includes both “In/Out” and “Surround” connectivity.
- Deleted elements may be “undeleted” until a new .inst file is written; deleted elements do not appear in output .inst or .geo files.
- Multiple geometry files can be generated with concurrent variation of up to 20 parameters interpolated either linearly or logarithmically between given limits.
- “Automatic drifts” between connected elements (if not already contiguous) are created *automatically* when the .geo files are written.
- All parts of the package are written in one language (Fortran), making updates easier.

Because of the last feature, NISP_Win can be used as a testbed for MC_Web (or as an alternative until we get support for updating MC_Web). Some revised and new elements are

- For a fixed source, may enter the Brightness and a Polarization vector.
- Fermi chopper: length of slit package and options for material types, including nuclear resonances (*cf.* Fig. 2).
- Cylindrical scattering chamber has option of beam axis rotation for exiting neutrons.
- Detectors (all): new options for Monitor and/or Polarization.
- Detectors (plane circular and plane rectangular): include origin of polar coordinates; encoding limits default to case size.
- For the Bender/Multichannel guide, axis rotation is optional.
- Magnetic Drift, with **B** interpolated between two Z-planes with preceding/following fringe regions.
- Magnetic induction region bounded by two Z-planes, for any number of current loops either within or exterior to the region (*e.g.*, Fig. 4).
- Magnetic induction for a solenoid (*e.g.*, Fig. 3). The geometrical region is a cylinder centered on the solenoid.

As always, any questions, comments, or requests should be addressed to PASeeger@aol.com and/or lld@lanl.gov. Including the word “NISP” in the subject is recommended. We urgently request new or improved algorithms. In particular we need more devices to manipulate or utilize neutron polarization.

Acknowledgments

This effort has been supported by the Los Alamos National Laboratory and the Manuel Lujan Jr. Neutron Scattering Center, a national user facility funded by the United States Department of Energy, Office of Basic Energy Sciences-Materials Science, under contract number W-7405-ENG-36 with the University of California.

References

- [1] P. A. Seeger, ICANS-XIII, Oct. 11-14, 1995, PSI Proceedings 95-02, 194 (1995).
- [2] T. G. Thelliez, L. L. Daemen, P. A. Seeger, and R. P. Hjelm, Jr., ICANS- XIII, Oct. 11-14, 1995, PSI Proceedings 95-02, 307 (1995).
- [3] P. A. Seeger, L. L. Daemen, R. P. Hjelm, Jr., and T. G. Thelliez, ICANS-XIV, June 14-19, 1998, ANL report ANL 98/33, vol. 1, 202 (1998).
- [4] M. W. Johnson and C. Stephanou, report RL-78-090 (1978); M. W. Johnson, report RL-80-065 (1980).
- [5] T. G. Thelliez, *thierry@acm.org*, private communication (2003).
- [6] P. A. Seeger, L. L. Daemen, T. G. Thelliez, and R. P. Hjelm, *Neutron News* **13** (2002) No. 4, p. 20.
- [7] P. A. Seeger, T. G. Thelliez, and L. L. Daemen, Los Alamos Report LAUR-00-1394, <ftp://strider.lansce.lanl.gov/pub/NISP/document/manual.pdf> (2000).
- [8] All source codes are available through the web site, <ftp://strider.lansce.lanl.gov/pub/NISP/fortran/>. If adapting to other contexts, please retain all authorship information.
- [9] J. E. Lynn and P. A. Seeger, *Atomic Data and Nuclear Data Tables* **44** (1990) 191.
- [10] V. F. Sears, *Acta Cryst.* **A53** (1997) 33; V. F. Sears, *Acta Cryst.* **A53** (1997) 46.
- [11] P. A. Seeger and L. L. Daemen, *Appl. Phys.* **A74** (2002) S1458.
- [12] P. A. Seeger, *Nucl. Inst. and Meth. Phys. Res. A* (accepted for publication, 2003).
- [13] P. A. Seeger and L. L. Daemen, *Nucl. Inst. and Meth. Phys. Res.* **A457** (2001) 338.
- [14] W. G. Williams, *Polarized Neutrons* (Clarendon Press, Oxford, 1988).
- [15] S. W. Lovesey, in *Theory of Thermal Neutron Scattering from Condensed Matter, vol. 2* (Clarendon Press, Oxford, 1984), Ch. 10.
- [16] E. Durand, *Magnetostatique* (Masson, Paris, 1968) pp.97-100.
- [17] J. Spanier and K. B. Oldham, *An Atlas of Functions* (Hemisphere, 1987), sec. 61–62.
- [18] Dimensions taken from <http://www.hmi.de/bensc/instrumente/v5/v5.html>.
- [19] P. A. Seeger, L. L. Daemen, E. Farhi, W.-T. Lee, X.-L. Wang, L. Passell, J. Šaroun, and G. Zsigmond, *Neutron News* **13** (2002) No. 4, p. 24.
- [20] S. M. Shapiro and N. J. Chesser, *Nucl. Inst. and Meth.* **101** (1973) 183.
- [21] ftp://strider.lansce.lanl.gov/pub/NISP/document/H8sim_Jan29.doc.
- [22] <ftp://strider.lansce.lanl.gov/pub/NISP/NISPforWindows.zip>.

Appendix A. Parameters of magnetic induction regions.

```
C class: magnetic field regions represented by induction vector B
C 1) first parameter of all B regions may be pulsed-source repetition
C time, or zero for a steady field (us)
C 2) pulse reference time T0, such that B(t) ~ 1/(t-T0) (us)
C 3) initial field time T1 measured from T0; B(T0+T1) = B from
C various algorithms and B(t) = B*T1/(t-T0) (us)
C 4) latest field time T2 measured from T0 (us)
C INTEGER B_PERIOD, B_T0, B_T1, B_T2
C PARAMETER (B_PERIOD=1,B_T0=2,B_T1=3,B_T2=4)
C type 60 = uniform field; 8 parameters
C 5-7) (X,Y,Z) components of magnetic induction, B1 (T)
C 8) |B1| (T)
C INTEGER B1x, B1y, B1z, ABSB_1, NUMBER_60
C PARAMETER (B1x=5,B1y=6,B1z=7,ABSB_1=8,NUMBER_60=9)
C type 61.n = gradient field between 2 surfaces; 39 parameters
```

```

C      .0 = linear interpolation of components of B
C      .1 = helical (linear magnitude and angle)
C  5-7) (X,Y,Z) components of magnetic induction at surface 1, B1 (T)
C  8)   |B1| (T)
C  9-11) (X,Y,Z) components of magnetic induction at surface 2, B2 (T)
C  12)  |B2| (T)
C  13-22) coefficients of surface 1, normalized to give distance in m
C  23-32) coefficients of surface 2, normalized to give distance in m
C  33-38) matrix elements for helical interpolation (/T)
C  39)  angle between B1 and B2 vectors (rad)
C      INTEGER B2x, B2y, B2z, ABSB_2, SURF1, SURF2, Bhelix, BlthetaB2, &
&          NUMBER_61
C      PARAMETER (B2x=9,B2y=10,B2z=11,ABSB_2=12,SURF1=13,SURF2=23, &
&          Bhelix=33,BlthetaB2=39,NUMBER_61=40)
C type 62 = superposition of current loops; 6 + 8*N parameters
C  5)   number of current loops
C  6)   number of parameters to describe each loop (8)
C  7-9) X,Y,Z of center of 1st loop (m)
C  10-12) direction cosines of 1st loop axis
C  13)  radius of 1st loop (m)
C  14)  current in 1st loop (A)
C  15-22) parameters of 2nd loop, etc.
C      INTEGER N_LOOPS, LOOPN, LOOPX, LOOPY, LOOPZ, LOOPALF, LOOPBET, &
&          LOOPGAM, LOOPRAD, LOOPCRNT, NUMBER_62
C      PARAMETER (N_LOOPS=5,LOOPN=6,LOOPX=7,LOOPY=8,LOOPZ=9,LOOPALF=10, &
&          LOOPBET=11,LOOPGAM=12,LOOPRAD=13,LOOPCRNT=14, &
&          NUMBER_62=7)
C type 63 = solenoid; 13 parameters
C  5-7) X,Y,Z at center of the solenoid (m)
C  8-10) direction cosines of the solenoid axis
C  11)  radius of the solenoid (m)
C  12)  half of the length of the solenoid (m)
C  13)  surface current density, turns*current/length (A/m)
C      INTEGER SOL_XCENT, SOL_YCENT, SOL_ZCENT, SOL_ALF, SOL_BET, &
&          SOL_GAM, SOL_R, SOL_HL, SOL_J, NUMBER_63
C      PARAMETER (SOL_XCENT=5,SOL_YCENT=6,SOL_ZCENT=7,SOL_ALF=8, &
&          SOL_BET=9,SOL_GAM=10,SOL_R=11,SOL_HL=12,SOL_J=13, &
&          NUMBER_63=14)
C type 64 = finite coaxial alternating current sheets; 16 parameters
C  5)   number of sheets
C  6)   spacing of sheets (m)
C  7-9) X,Y,Z of center of 1st sheet (m)
C  10-12) direction cosines of sheet normal (axis of array)
C  13-14) transverse dimensions of sheets (m)
C  15)  sheet current (A)
C  16)  azimuthal angle of current in 1st sheet (rad)
C      INTEGER N_SHEETS, SHT_SPACE, SHT_XCENT, SHT_YCENT, SHT_ZCENT, &
&          SHT_ALF, SHT_BET, SHT_GAM, SHT_DX, SHT_DY, SHT_I, &
&          SHT_PHI, NUMBER_64
C      PARAMETER (N_SHEETS=5,SHT_SPACE=6,SHT_XCENT=7,SHT_YCENT=8, &
&          SHT_ZCENT=9,SHT_ALF=10,SHT_BET=11,SHT_GAM=12, &
&          SHT_DX=13,SHT_DY=14,SHT_I=15,SHT_PHI=16,NUMBER_64=17)
C type 65 = field defined in file; 4 parameters
C      INTEGER NUMBER_65
C      PARAMETER (NUMBER_65=5)
C type 69 = superposition of any number of field types; 5+n parameters
C  5)   number of superposed fields
C  6...) offset to parameter block with field definition
C      INTEGER N_FIELDS, B_OFFSET, NUMBER_69
C      PARAMETER (N_FIELDS=5,B_OFFSET=6,NUMBER_69=7)

```