CONF-9510212--1

LA-UR- 95-2937

Title: THE MCLIB LIBRARY: MONTE CARLO SIMULATION OF NEUTRON SCATTERING INSTRUMENTS

Author(s): P. A. Seeger

Submitted to: ICANS XIII, Villigen, Switzerland, 11-14 Oct 95

Los Alamos
NATIONAL LABORATORY

Form No. 836 R5
ST 2629 10/91

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ICANS-XIII
13th Meeting of the International Collaboration on
Advanced Neutron Sources
October 11-14, 1995
Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

THE MCLIB LIBRARY: MONTE CARLO SIMULATION OF NEUTRON SCATTERING INSTRUMENTS

Philip A. Seeger

239 Loma del Escolar, Los Alamos, NM 87544, U.S.A.
E-mail: PASeeger@aol.com

ABSTRACT

This report describes the philosophy and structure of MCLIB, a Fortran library of Monte Carlo subroutines which has been developed for design of neutron scattering instruments. A pair of programs (LQDGEOM and MC_RUN) which use the library are shown as an example.

1. Introduction

Monte Carlo is a method to integrate over a large number of variables. Random numbers are used to select a value for each variable, and the integrand is evaluated. The process is repeated a large number of times and the resulting values are averaged. For a neutron transport problem, we first select a neutron from the source distribution, and project it through the instrument using either deterministic or probabilistic algorithms to describe its interaction whenever it hits something, and then (if it hits the detector) tally it in a histogram representing where and when it was detected. This is intended to simulate the process of running an actual experiment (but it is *much* slower).

The present MCLIB library has been derived from codes written by Mike Johnson at the Rutherford Laboratory [1]. Significant additions and revisions were made by this author in 1984 [2], and the entire code was rewritten in a structured form in 1994 [3]. Whenever the code has been applied to new problems, additions have been made. Thus significant contributions to the present library have been made by Richard Heenan (Rutherford-Appleton Laboratory), and by Glenn Olah, Bob VonDreele, Greg Smith, and Luke Daemen (Los Alamos neutron Science Center, LANSCE). Mike Fitzsimmons and Joyce Goldstone have contributed greatly to the debugging process. Several applications of the code were presented at an instrument design workshop [4]. A new collaboration with Larry Passell and Uli Wildgruber (Brookhaven National Laboratory) is also producing improvements.

The process is carried out in two stages. First a program must be generated to describe the geometry of the specific instrument being simulated; for example, the program LQDGEOM may be used to define a small-angle scattering instrument with pinhole collimation, up to three

choppers, and an on-axis 2-dimensional position sensitive detector. Essentially all of the user interaction occurs in this stage. The output is a geometry file containing the complete problem definition. That file is then passed to a second-stage program, for example MC_RUN, which transports neutrons and tallies results in histograms. Principal outputs are a file with a statistical summary, and a data file with histograms of the spectrum and detector. A third stage, which is not part of the Monte Carlo process, is to perform whatever data reduction is appropriate to the experiment being simulated. Some measure of the information content (or a "figure of merit") is then used to evaluate the design of the instrument.

Features of MCLIB which are different from other Monte Carlo libraries include

- Simplified transmission through materials. Rather than compute microscopic interaction in a simple (amorphous unpolarized) region, attenuation of the transmitted neutron is calculated.
- Optics at surfaces. When a neutron reaches a surface, the (complex) index of refraction is computed to decide whether the neutron will reflect or refract.
- Time-dependent devices. There are element types to describe moving devices such as choppers or a gravity focuser.
- Acceleration of gravity is included in transport.
- Scattering functions. Each kind of scattering sample is an element type. The scattering algorithm may be deterministic (reflectometry), probabilistic (hard-sphere scatterer), or a combination (Bragg reflection into a Debye-Scherrer cone).

2. Geometry Description by Surfaces and Regions

The geometry of a system is described by surfaces and regions. A *surface* is defined by a general 3-dimensional quadratic equation of the form

$$A\,x^2 + B\,x + C\,y^2 + D\,y + E\,z^2 + F\,z + G + P\,xy + Q\,yz + R\,zx = 0 \qquad (1)$$

with 10 coefficients, plus a roughness parameter (*BETA*). The surface divides 3-dimensional space into two parts, which are called the + and – sides of the surface depending on whether the left-hand side of eq. (1) evaluates to a positive or a negative value. For example, a plane perpendicular to the z-axis at z = 1 can be expressed by the equation

$$z - 1 = 0,$$

i.e., $F = 1$ and $G = -1$ (all other coefficients zero). Then all points with $z < 1$ are on the – side and all points with $z > 1$ are on the + side of the surface. The scaling of eq. (1) is arbitrary, but we tend to evaluate it as m^2, so that coefficients *A*, *C*, *E*, *P*, *Q*, and *R* are dimensionless, coefficients *B*, *D*, and *F* are in m, and *G* is m^2. The parameter *BETA* is the length of a randomly oriented 3-dimensional vector which is added to the unit vector normal to the mathematical surface to determine the surface orientation when a particle interacts. For a perfect smooth surface, $BETA = 0$; for $0 < BETA < 1$, *BETA* is the sine of the maximum angular deviation of the surface normal from smooth. If $BETA < 0$ (or $BETA \gg 1$), the surface is completely random.

Note that the coordinate system being used is left handed: the instrument axis is the positive z-direction, the x-axis is horizontal and positive to the right, and the y-axis is vertical with the acceleration of gravity in the negative y-direction.

The geometric shape of each *region* is defined by its relationship to all of the defined surfaces. A positive or negative integer is placed in the region definition if every point in the region is on the + or – side of the corresponding surface, and surfaces which do not bound the region are set to zero. Special characteristics of the boundary are given by the value of the integer: ±1 for an ordinary surface with roughness *BETA* and the possibility of refraction or critical reflection; ±2 for total reflection; ±3 for diffuse scattering (independent of *BETA*); ±4 for total absorption; and ±5 for cases requiring special action (such as a coordinate transformation) whenever a particle enters or leaves the region. Generally, no surface may be used as a boundary of a region if any part of that surface is inside the region, because then some points in the region would be on the + side and some on the – side of that surface. Concave (reentrant) shapes are allowed when using quadratic surfaces as in fig. 1a, but the shape in fig. 1b requires that two regions be defined. Another method to define a reentrant region is to exclude embedded regions, which is accomplished by adding 10 to the surface type number in the definition of the enclosing region (*e.g.*, "4" becomes "14" or "–1" becomes "–11"). Such surfaces are *not* tested as part of the definition of being inside the region. When the trajectory of a particle inside the region intersects the surface, it will exit if a valid region exists on the other side, but will otherwise remain in the enclosing region. This method must be used with care, since particles within the embedded regions also pass the test for being within the enclosing region.

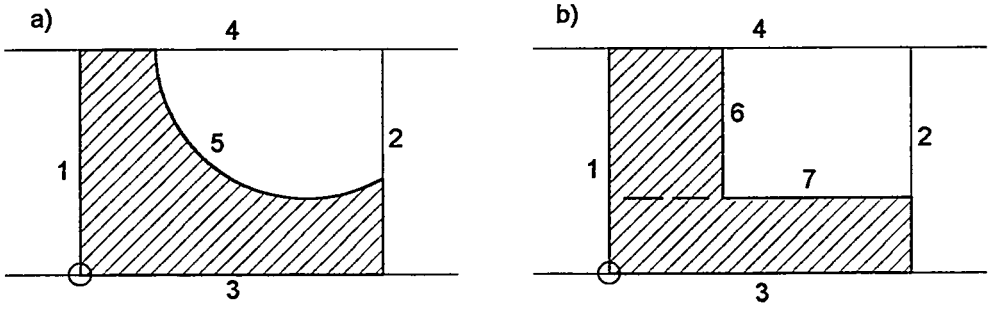

Figure 1. Concave regions.

Equations of the numbered surfaces are

1:		x				$= 0$			
2:		x			-4	$= 0$			
3:				y		$= 0$			
4:				y	-3	$= 0$			
5:	x^2	$-6x$	$+y^2$	$-6y$	$+14$	$= 0$	or	$(x-3)^2 + (y-3)^2 = r^2 = 4$	
6:		x			-1.5	$= 0$			
7:				y	-1	$= 0$			

The shaded area in a) can be expressed unequivocally as a single region:

+1 −1 +1 −1 +1 0 0

To avoid ambiguity, the area in b) requires two regions, divided as shown by surface 7 (or alternatively by surface 6).

+1 −1 +1 0 0 0 −1
+1 0 0 −1 0 −1 +1

The unshaded area in either case requires 5 regions for a complete unambiguous definition. (One choice of region boundaries is shown by the faint lines.) If the upper right corner

0 −1 0 −1 0 +1 +1

is considered an embedded region, then the shaded area may also be defined as

+1 −1 +1 −1 0 −11 −11

3. Instrument Elements

Each region is also an *element* of the instrument. Every element has a *NAME* and a pointer (*INDEX*). For a void drift region, *INDEX* = 0, and for every element which is not a void

INDEX points to a location in a REAL*4 parameter block which contains the element type number, possibly followed by additional parameters. Future development of the library should be accomplished by defining new element types and implementing the corresponding algorithms for how a neutron interacts in such regions. Presently defined types and their parameters are listed below; file MC_ELMNT.INC (Appendix A) includes definitions of mnemonic symbols for all the parameters, with values equal to the address offset from the *INDEX* pointer.

type 0 = total absorber; no parameters
type 1 = amorphous unpolarized material; 4 parameters
Real and Imaginary scattering-length density (10^{10}/cm^2)
macroscopic scattering cross section (1/m)
velocity-dependent cross section, at 1 m/μs (1/μs)
type 2 = aluminum, including Bragg edges; no parameters
type 3 = hydrogenous, including multiple scattering
type 4 = supermirror characterized by 2 scattering densities; 5 parameters
Complex bulk material scattering-length density (10^{10}/cm^2)
Complex effective supermirror density (10^{10}/cm^2)
Relative efficiency of supermirror regime
type 5 = beryllium at 100K, including Bragg edges; no parameters
type 6 = single-crystal filter, Freund formalism; 3 parameters
$\Sigma = \Sigma_{free} \times \{1 - \exp(-C_2/\lambda^2)\} + \Sigma_{abs} \times \lambda$
Σ_{free} = limiting (short wavelength) free-atom macroscopic cross section (1/cm)
$C_2 = -\ln\{1 - (\Sigma(1\text{Å}) - \Sigma_{abs})/(\Sigma_{free} - \Sigma_{abs})\}$ (Å^2)
Σ_{abs} = sum of 1/v macroscopic cross sections at 1Å (1/cm/Å)
type 10 = multi-aperture collimator
type 11 = multi-slit collimator, vertical blades; 3 or 5 parameters
Note: the type 11 region must be followed immediately by up to 5 additional regions defining the center passage and the bounding blades, and the entrance/exit surfaces must be surface type 5.
Spacing of slits, centerline-to-centerline (m)
Rate of convergence (>0) or divergence (<0) of one slit
Z at entrance of the region, where spacing is measured (m)
For a curved system (bender),
sine of half the angle of bend
cosine of half the angle of bend
type 12 = multi-slit collimator, horizontal blades; 3 or 5 parameters
Same parameters as type 11 (see *Note* under type 11)
type 13 = crystal monochrometer; 10 parameters
Twice the crystal plane spacing (Å)
Nominal Z position for rotation of instrument axis (m)
Sine and cosine of take-off angle
X-, Y-, and Z-components of mosaic spread, rms of sines of angles
rms spread of plane spacing, σ_d/d
max number of loops (or microcrystal orientations) to try
probability normalization factor per try, derived from reflection probability at peak wavelength: $1 - (1 - \text{max\_prob})^{(1/\text{trys})}$
types 20.n = chopper (disk or blade); 6 parameters

20.2 or 20.3 is counter-rotating (fully closed when edges at 0)
Linear velocity of opening crossing beam centerline (m/μs)
Time to cover or uncover half the width of the moderator (μs)
Nominal time at which opening chopper edge crosses zero (μs)
Nominal time at which closing chopper edge crosses zero (μs)
Phase jitter of chopper, rms (μs)
Period of chopper (μs)

type 21 = Fermi chopper (not implemented)

type 22 = gravity focuser; 5 parameters
Note: the type 22 region must be followed immediately by 2 additional regions defining the aperture, and the entrance/exit surfaces must be surface type 5.
acceleration (m/μs^2), and rms phase jitter (μs)
nominal times for start and top of upward stroke (μs)
time between pulses (μs)

type 23 = removable beamstop; no parameters

type 30 = sample which scatters at constant Q; 2 parameters
–ln(transmission at 1 Å)
value of Q for scatter (1/Å)

type 31 = scattering sample of hard spheres; 2 parameters
–ln(transmission at 1 Å)
hard-sphere radius for scatter (Å)

type 32 = isotropic scatterer with fixed energy change; 2 parameters
–ln(transmission at 1 Å)
inelastic energy change (0 if elastic) (meV)

type 34 = inelastic scattering kernel; no parameters; *NAME* is '[path]filename' of $S(\alpha,\beta)$ file in MCNP Type I format

type 35 = scattering from layered reflectometry sample; 1 + 4 *N* parameters
number of layers, including substrate
parameters for each layer, starting with substrate:
4π x Real and Imaginary scattering-length density (1/Å^2)
Thickness of layer (zero for substrate) (Å)
Roughness, 2 σ^2 of outer surface of this layer (Å^2)

type 36 = scattering from isotropic polycrystaline powder; 6 parameters + 2 × table length
number of Bragg edges included
limiting (short wavelength) macroscopic total xsection (1/cm)
macroscopic incoherent scattering cross section (1/cm)
macroscopic 1/v scattering cross section at 1 Å (1/cm/Å)
macroscopic 1/v absorption cross section at 1 Å (1/cm/Å)
table of d-spacings of Bragg edges (Å), followed by explicit 0 and table of cummulative macroscopic cross sections at 1 Å (1/cm/Å^2)

type 40 = detector; 9 parameters
surface number
–ln(1 – efficiency at 1 Å)
time-of-flight clock parameters:
minimum and maximum times (μs)
number of time channels
if logarithmic, dt/t (otherwise dt/t = 0)
minimum clock tick in determining log scale (μs)

electronic delay of detector events (μs)
repeat period of data-acquisition electronics (μs)
Note: if t-o-f is logarithmic, *TMAX* is overridden

type 41 = vertical linear detector; 14 parameters
surface number
–ln(1 – efficiency at 1 Å)
time-of-flight parameters (7)
locations of bottom and top of detector (m)
number of detector elements
size of detector element (m)
root-mean-square encoding error of detector (m)

type 42 = horizontal linear detector; 14 parameters
surface number
–ln(1 – efficiency at 1 Å)
time-of-flight parameters (7)
locations of left and right ends of detector (m)
number of detector elements
size of detector element (m)
root-mean-square encoding error of detector (m)

type 43 = 2-D rectilinear detector; 18 parameters
surface number
–ln(1 – efficiency at 1 Å)
time-of-flight parameters (7)
locations of left and right edges of detector (m)
number of detector elements in the horizontal direction
width of detector element (m)
root-mean-square X encoding error of detector (m)
locations of bottom and top edges of detector (m)
number of detector elements in the vertical direction
height of detector element (m)
root-mean-square Y encoding error of detector (m)

type 44 = longitudinal linear detector; 14 parameters
surface number
–ln(1 – efficiency at 1 Å)
time-of-flight parameters (7)
locations of upstream and downstream ends of detector (m)
number of detector elements
size of detector element (m)
root-mean-square encoding error of detector (m)

type 50 = scattering chamber, void-filled. No parameters, but other regions may be embedded, indicated by surface types with 10s digit on.

types 90.n = source size and phase space to be sampled; 14-18 parameters
edges of rectangular moderator face (m)
location and radius (half-width) of apertures which define beam (m)
additional vertical space to sample for gravity focus (m)
min and max neutron energy to be sampled (eV)
time between beam pulses and square pulse width (μs)
offset to parameter block with spectrum and line shape parameters

(optional) vertical offsets of apertures, type 90.4 (m)

type 91 = source energy distribution table and line shape parameters; 12 parameters plus length of table
number of entries in energy table (1 is special case)
location in table of center of normal distribution (index units)
or value of nominal neutron velocity (m/μs) (if number of entries = 1)
standard deviation of normal distribution (table index units)
or relative fwhm of velocity selector (if number of entries = 1)
source brightness, summed over limits of *E_TABLE* (n/ster/m^2/MW/s)
1 or 2 exponential time constants in thermal (low-energy) limit (μs)
probability of 2nd exponential
epithermal (high energy) time constant proportional to lambda (μs/A)
Gaussian delay and width proportional to λ (μs/Å)
switching function 1/e point (Å) and power (slope)
origin of table of cumulative energy distribution (weighted by λ^2) of source spectrum on equally spaced normal-curve values of log(E / 1 meV)

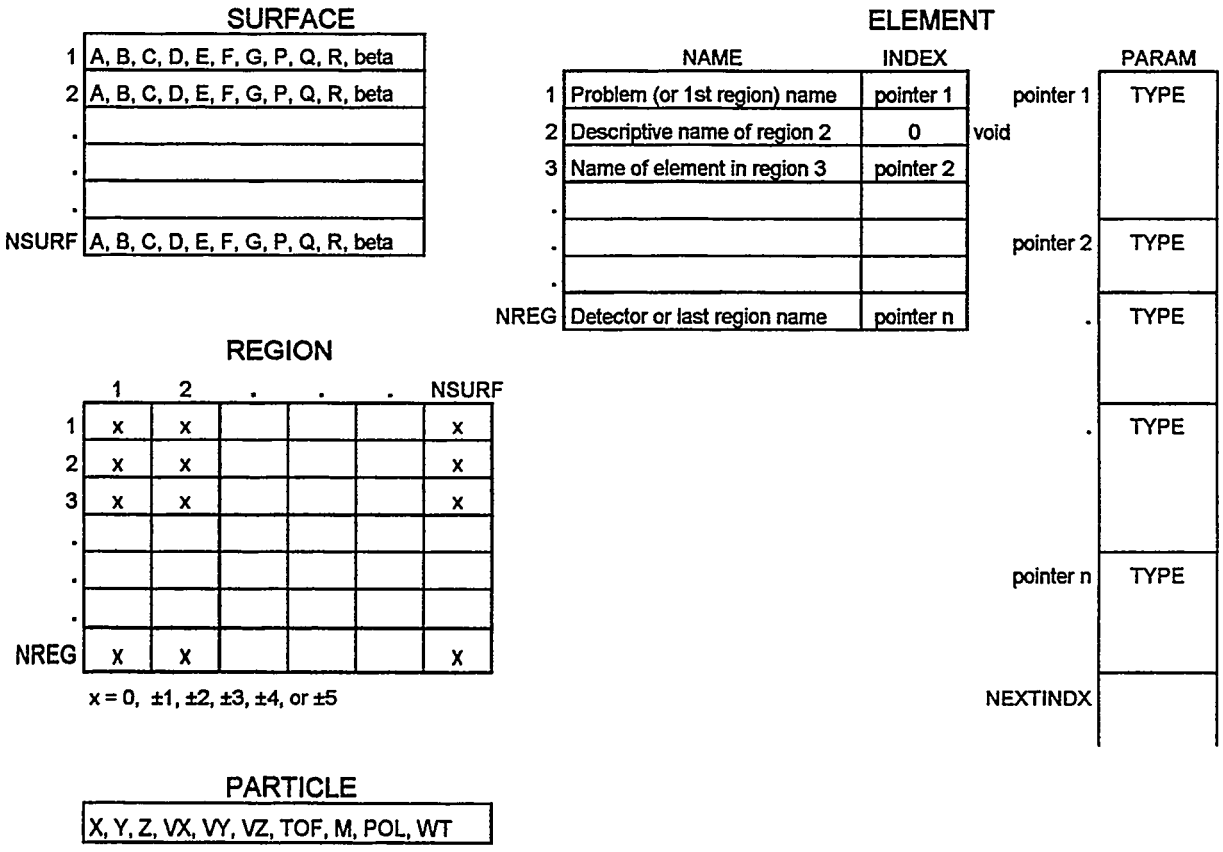

Figure 2. Structures used in MCLIB.

4. Program Structures

The relationships of the structures are shown schematically in fig. 2. The library source code is available in Fortran 77 with VAX structure extensions (F77/VAX), and also in Fortran 90 (F90). Unfortunately the F90 syntax did not adopt the relatively widespread F77/VAX standard, most distressingly by using "%" instead of "." as the member pointer in a structure! In F77/VAX, each surface is a `RECORD` of type `/SURFACE/`, and elements are referenced as

(e.g.) SURFACE.G. In F90, the surfaces are defined as derived TYPE (SURFACE), and element references are of the form SURFACE%G. Similarly, a geometric region is a RECORD of type /REGION/ or a derived TYPE (REGION); the structure in either case is a vector *IGEOM* of 2-byte integers, of length equal to the maximum allowed number of surfaces. An additional structure, MC_GEOM, contains the numbers of surfaces and regions in the problem, *NSURF* and *NREG*, and arrays of surface and region structures. Information about elements is contained in a structure called MC_ELEMENT which includes *NAME* and *INDEX* arrays, the parameter block *PARAM*, and the pointer *NEXTINDX* to the next available location in *PARAM*.

The final structure is PARTICLE, which includes the position, velocity, time of flight, mass (1 for a neutron), polarization (not yet implemented in the code), and statistical weight of a particle. A purely "analog" Monte Carlo traces each individual neutron until it is either lost or detected. MCLIB uses "weighted" neutrons, and in many of the processes the statistical weight is multiplied by the probability of survival instead of using a random number to decide whether to terminate the history ("Russian Roulette"). This is especially beneficial when scattering probability is small, as in subcritical reflection. To track more long-wavelength neutrons (which in general have larger scattering probability), the source distribution used is $\lambda^2 I(\lambda)$ instead of $I(\lambda)$ and the initial weight is proportional to $1/\lambda^2$. The tallied results are then the sum of detected neutron weights. The relative error in each bin, however, depends on the number of histories recorded.

The F77/VAX and F90 structure definitions follow. As compiled, the dimensions of the arrays are *MAXS* = 150, *MAXR* = 100, and *MAXP* = 800. For use in coding, all structure definitions are included in file MC_GEOM.INC (Appendix A).

F77 / VAX

```
STRUCTURE /SURFACE/
   REAL*4 A,B,C,D,E,F,G,P,Q,R
   REAL*4 BETA
END STRUCTURE

STRUCTURE /REGION/
   INTEGER*2 IGEOM(MAXS)
END STRUCTURE

STRUCTURE /MC_ELEMENT/
   CHARACTER NAME(MAXR)*40
   INTEGER*4 NEXTINDX, INDEX(MAXR)
   REAL*4    PARAM(MAXP)
END STRUCTURE

STRUCTURE /MC_GEOM/
   INTEGER*4 NSURF,NREG
   RECORD /SURFACE/ SURFACE(MAXS)
   RECORD /REGION/  REGION(MAXR)
END STRUCTURE

STRUCTURE /PARTICLE/
   REAL*4 X, Y, Z, VX, VY, VZ
   REAL*4 TOF, M, POL, WT
END STRUCTURE
```

F90

```
TYPE SURFACE
   SEQUENCE
   REAL 4:: A,B,C,D,E,F,G,P,Q,R
   REAL 4:: BETA
END TYPE SURFACE

TYPE REGION
   INTEGER 2:: IGEOM(MAXS)
END TYPE REGION

TYPE MC_ELEMENT
   SEQUENCE
   CHARACTER   NAME(MAXR)*40
   INTEGER 4:: NEXTINDX, INDEX(MAXR)
   REAL 4::    PARAM(MAXP)
END STRUCTURE

TYPE MC_GEOM
   SEQUENCE
   INTEGER 4:: NSURF,NREG
   TYPE (SURFACE) SURFACE(MAXS)
   TYPE (REGION)  REGION(MAXR)
END TYPE MC_GEOM

TYPE PARTICLE
   SEQUENCE
   REAL 4:: X, Y, Z, VX, VY, VZ
   REAL 4:: TOF, M, POL, WT
END TYPE PARTICLE
```

5. Subroutine Library

Complete descriptions, calling sequences, revision history, and externals of all of the library subroutines are given in Appendix B. The following listing divides the subroutines into (somewhat arbitrary) categories, and includes the latest modification date and an abbreviated description. Note that if the operating system does not include the function RAN(ISEED), then the function must be provided as part of the library.

Vector analysis and transport

ANGLI	03 Feb 94	angle of incidence
DIST	23 Mar 95	distance to a surface
DTOEX	06 Mar 95	distance to nearest boundary
ELSCAT	26 Aug 95	do elastic scattering
LIGHTRFL	19 Mar 85	light reflection probability
LMONOCRM	08 Jun 95	reflection by crystal monochromator
LREFLCT	07 Jan 85	neutron reflection probability
MOSAIC	27 Jan 95	angle of incidence with mosaic spread
MOVEX	19 Apr 94	move a particle
NEXTRG	29 Jun 95	find region across boundary
N_SOURCE	28 Mar 95	get source neutron
GET_SPACE		get phase space of source
OPERATE	26 Aug 95	find what happens to particle within region
EXIT_REG		find what happens to particle leaving region
OPTICS	03 Feb 94	photon at region boundary
RFLN	03 Feb 94	do reflection
SNELL	03 Feb 94	apply Snell's law
TESTIN	29 Jun 95	find if within region
WHICHR	03 Feb 94	find what region particle is in
WOBBLE	03 Feb 94	angle of incidence at wavy surface

Material properties and scattering functions

ATTEN_Al	28 Jan 93	attenuation of Al
ATTEN_Be	25 Jan 95	attenuation of Be
ATTEN_X	14 Aug 95	attenuation of single-crystal filter
KERNEL	03 Apr 95	inelastic scattering kernel
PLQSPHR	06 Mar 95	probability from spherical scatterer
POWDER	14 Jul 95	scatter from polycrystalline powder
REFLAYER	07 Feb 95	reflection probability from multiple layers

Random distributions

ORRAND	11 Mar 95	random unit vector
PLCNVL	11 Mar 95	probability from convolution of top-hats
PLEXP	16 Feb 95	probability from exponential
PLNORM	11 Mar 95	probability from normal
PLNRMTBL	13 Feb 95	probability from table vs. normal distribution
PLPOISSN	18 Mar 95	probability from Poisson
PLTIME	05 Sep 95	probability from parametrized emission time
PLTRNGL	04 Jan 85	probability from triangle
RAN	various	random number; platform dependent
RAN0	86	desequentialized random number
RNDCRCL	11 Mar 95	random point in unit circle

General utilities

DIGITS	05 Dec 87	convert number to string
GAMMLN	92	ln(Γ(x))
GET_BINS	26 Jul 95	find detector/time bins
GET_RHO	05 Jan 95	find scattering-length density
GINI	05 Feb 92	standard deviation by Gini statistic
HUNT	13 Apr 87	find index in array
INTERP	84	interpolate in array
NORM	78	normalize unit vector
READ_1D	14 Aug 92	read 1-D block ASCII data file
READ_2D	06 Apr 95	read 2-D block ASCII data file
REALOUT	16 Aug 95	output array of REAL*4 numbers in block ASCII

Time-dependent devices

GRAV_FOC	07 Apr 94	gravity focuser
XCHOPPER	15 Mar 93	disk or blade chopper

Subroutine OPERATE and its other entry point EXIT_REG play special roles in the transport process. As can be inferred from the abstract in Appendix B, if this subroutine is called when a neutron enters a new region, then the routine will determine what happens. Possible results on exit from OPERATE include detection or loss of the neutron, exiting the region with reduced statistical weight (partial absorption), splitting the neutron into a transmitted and a scattered particle with the sum of statistical weights equal to the original weight, or a transform of the coordinate system of the problem. Coordinate transforms are applied for element types made up of sub-regions (*e.g.*, Soller slits and benders), elements which change the beam direction (benders and monochromaters), and time-dependent devices (gravity focuser). In order to assure that the coordinate system is properly restored when the neutron leaves the region, EXIT_REG must be called; this is flagged by using ±5 as the integer defining all exit surfaces from such a region.

6. Neutron Source Functions

Subroutines NSOURCE and PLTIME are called to generate source neutrons, using table lookup to select the velocity and a parametrized model to select the emission time. (If the length of the lookup table is 1, the velocity is a triangular or delta-function distribution; if the thermal decay constant is ≤ 0, emission time is 0.) The table and parameters are generally read from a file derived either from experimental measurements or from a detailed Monte Carlo simulation of a target/moderator/reflector system. The example given below uses a coupled liquid H_2 moderator with a 60-cm thick Be reflector on a spallation source, computed with MCNP [5], but the procedure would be the same for an experimental spectrum measurement. The energy spectrum from the MCNP output is shown in fig. 3. Program MKNRMTBL is provided to convert the spectrum. To improve sampling at long wavelengths, the data are multiplied by λ^2 before the cumulative sum is formed (the neutron is given an initial statistical weight proportional to $1/\lambda^2$ to account for this). A Gaussian (*vs.* log E) is fitted to the weighted data, and the cumulative sum is compared to the integral of that Gaussian to form the table; that is, for equal steps in the argument of the Gaussian, its integral is evaluated and the cumulative sum is interpolated to find the corresponding neutron energy (tabulated as $\log_{10}(E$ / 1 meV). To sample the neutron energy, a random deviate is chosen from a Gaussian distribution and that value is used to interpolate in the table.

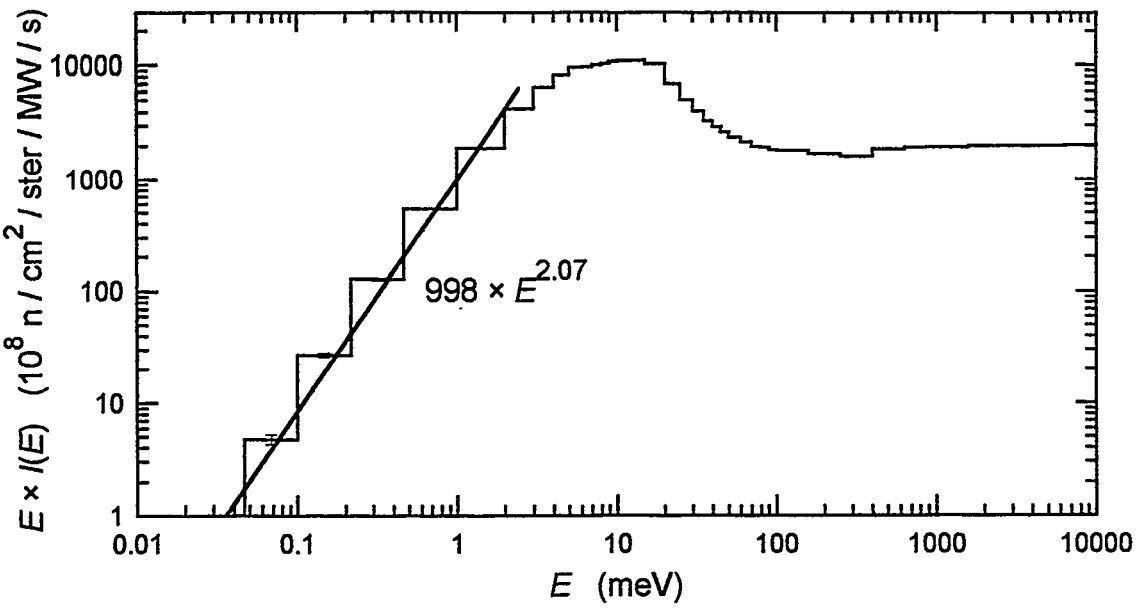

Figure 3. Neutron source energy distribution, as computed with the MCNP code.
Coupled liquid hydrogen moderator (equal fractions ortho and para), 40-cm Be (with D_2O) reflector, flux-trap geometry. A power law was fitted to the low-energy regime to interpolate the histogram.

The model for the time distribution is the sum of an epithermal and a thermal term convoluted with a Gaussian, added using a switch function [6,7] of the form $\exp[-(\lambda_0/\lambda)^P]$. This parametrization was chosen because the convolution is easy in Monte Carlo. Adequate fits can be made to most source terms using two exponentials; the epithermal (and the two parameters of the Gaussian) are proportional to wavelength, and the thermal time constant is independent of wavelength. Including the switch function, the total number of parameters in the model is then six. For the hybrid Be/Pb reflector case shown here, however, the fit with just two exponentials is not adequate and the thermal term must be a (constant) linear combination of two exponentials, for a total of eight parameters. Neutrons escaping from the moderator surface in the MCNP run were tallied *vs.* time for 18 (logarithmic) wavelength bins from 0.09 to 15 Å; fig. 4 shows the data for one such bin, from 0.9 to 1.2 Å. The exponential decay after 1000 μs being constant for all wavelengths, the data from 0.9 to 15 Å were summed and a least-squares fit was made over the time range 1000– 7000 μs, giving a time constant of 615 μs. Since the early (epithermal) parameters are all proportional to λ, the early-time data can be summed after binning *vs.* (t/λ). (The variable t/λ is equivalent to the reduced time vt used in ref. [6].) A least-squares fit for t/λ between 1 and 30 μs/Å gave the values epithermal exponential = 9.26 μs/Å, delay = 3.57 μs/Å, and width = 2.17 μs/Å. Given these fits to the early and late times, the additional exponential term was found (τ = 170 μs, ratio = 2/3). Then the ratios of thermal to epithermal terms were extracted, and the two parameters of the switch function determined (λ_0 = 1.05 Å, P = 2.345). The line in fig. 4 is an example of the resulting fit, evaluated at λ = 1.0 Å where all three components are significant.

7. Sample Program Output

Sample program source codes `LQDGEOM.FOR` and `MC_RUN.FOR` will be distributed with all copies of the library, but because of their lengths are not included in this report. Communication between the two programs is through a geometry file, such as the following example, `LPSS.GEO`. The first part of the file is not used by the succeeding program, but contains text to identify the problem and to give the parameters entered by the user.

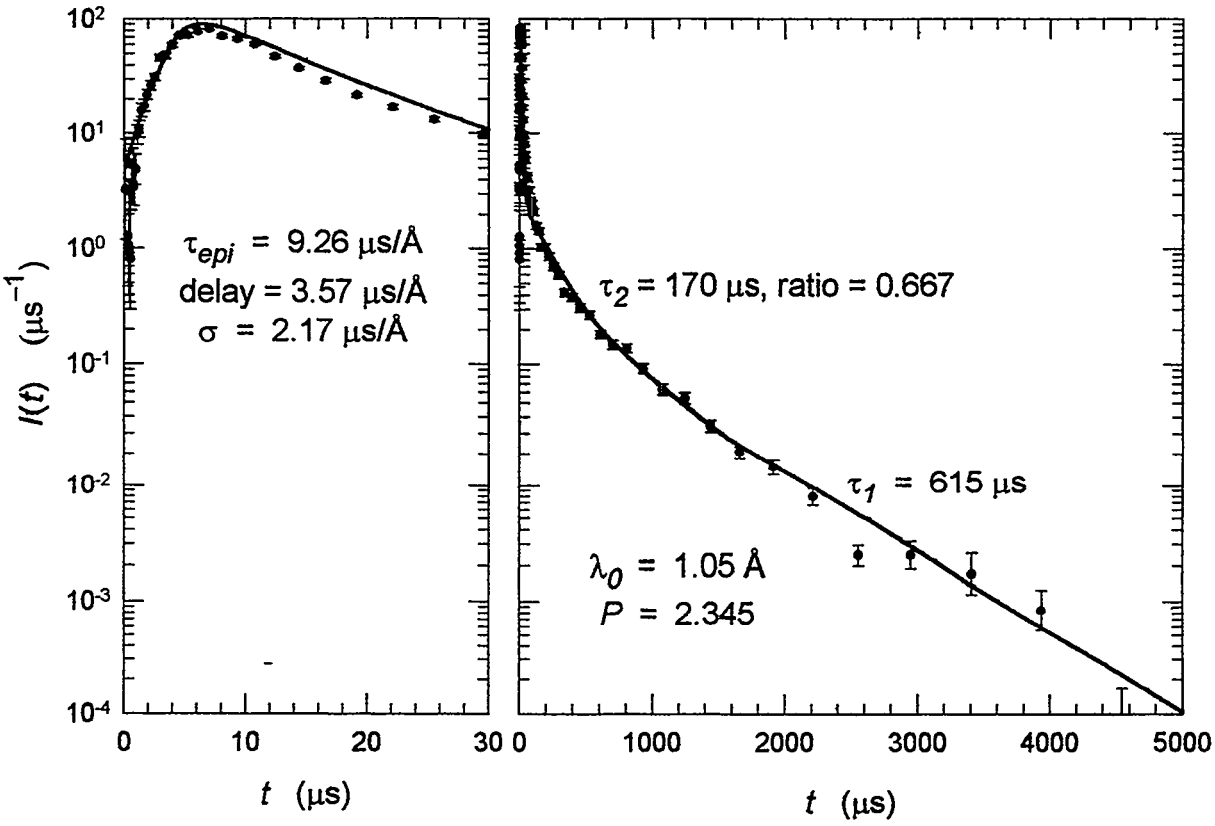

Figure 4. Source time distribution at 1.0 Å.
Points are the MCNP "data," and the line is the model fit. At this wavelength, near the midpoint of the switch function, the early "epithermal" and the two late "thermal" exponential terms are all seen.

```
Geometry code version: F90, June 30, 1995, P.A.Seeger
LQD Geometry file: lpss.geo
Proton pulse rate = 60.0 Hz, width = 1000.0 us, max wavelength 15.1 A
Sample at  7.00 m, Detector at 13.00 m from moderator
Sample diameter = 10.00 mm
Spectrum and lineshape file: \mclib\h240bepb\h240bepb.tbl
   T0 Chop at  2.60m, open  0.827ms, close  14.839ms, rms  40.0us, v 56.55m/s
O'lap Chop at  3.10m, open  7.257ms, close  12.281ms, rms  20.0us, v 94.25m/s
Frame Chop at  6.00m, open 16.253ms, close  22.874ms, rms  20.0us, v 94.25m/s
Collimation:  entrance aperture radius  7.53 mm at position  2.500 m
              exit                      4.44 mm              6.800 m
Moderator half-width and half-height = 0.050 m
          spot radius  9.32 mm, penumbra radius 14.48 mm
Time range  34370.0 -  49665.1 us
Sample Transmission at Lmax (15.11 A) is 0.600
Total thickness of Al = 12.0 mm
Detector half-width 0.320 m,  pixel 5.00 mm,  rms 3.40 mm
         penumbra radius 21.70 mm,  min angle 4.78 mr
Q-range:  0.00200 - 0.03134 /A
Gravitational droop  2.21 -   4.62 mm
```

The next three lines give the numbers of surfaces and regions and the length of the parameter block. They are required, and the remainder of the file must follow in order:

```
SURFACES      25
REGIONS       28
PARAMETERS   181
```

Each surface definition is given on one line; zero entries may be left null:

,,,,,1/	PZ (plane perpendicular to z-axis) at z = 0
1,,1,,,,-.0036/	CZ (cylinder on z-axis), radius 60 mm
,,,,,1,-2.498/	PZ at z = 2.498 m
486749,,486749,,-1,15.5,-60.0625/	Cone on z-axis
,,,,,1,-2.5/	PZ at z = 2.5 m
,,,,,1,-2.6/	PZ at z = 2.6 m
1,,1,0.6,,,-.2025/	Cylinder parallel to z-axis, offset y = –0.3 m, radius 0.335 m
,,,,,1,-2.9/	PZ at z = 2.9 m
1,,1,,,,-.0036/	CZ, radius 60 mm
,,,,,1,-3.1/	PZ at z = 3.1 m
1,,1,0.6,,,-.2025/	Cylinder parallel to z-axis, offset y = –0.3 m, radius 0.335 m
,,,,,1,-3.102/	PZ at z = 3.102 m
,,,,,1,-6/	PZ at z = 6.0 m
1,,1,0.6,,,-.2025/	Cylinder parallel to z-axis, offset y = –0.3 m, radius 0.335 m
,,,,,1,-6.002/	PZ at z = 6.002 m
,,,,,1,-6.8/	PZ at z = 6.8 m
585518,,585518,,-1,6.8,-11.56/	Cone on z-axis
,,,,,1,-6.802/	PZ at z = 6.802 m
,,,,,1,-7/	PZ at z = 7.0 m
,,,,,1,-7.001/	PZ at z = 7.001 m
,,,,,1,-7.013/	PZ at z = 7.013 m
1,,1,,,,-0.25/	CZ, radius 0.5 m
,,,,,1,-13/	PZ at z = 13.0 m
,,,,,1,-12.95/	PZ at z = 12.95 m
1,,1,,,,-5.6169-4/	CZ, radius 23.7 mm

Each region is a line with an entry for each surface:

```
-1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
-1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 1 -4 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  4 -4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  1 -4 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  1  4 -4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  1 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  1 -4 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  1  4 -4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  1 -4 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  1  4 -4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  1 -4 -1  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  1  4 -4  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0 -4  0  0  1 -1  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  4  0  0  1 -4  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  1 -4 -1  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  1  4 -4  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0 -4  0  0  0  0  0  1 -1  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  4  0  0  0  0  0  1 -4  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 -4 -1  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  4 -1  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 -4 -1 -11 14
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  4 -4  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  1 -4
```

The *NAME* associated with each region; the first *NAME* is the problem title, and the second is the title of the source file:

```
 7m, 13m, 15A, 60Hz (1000us), Q=0.01/A
H2 (50/50,64/16), Fluxtrap, 40-cm Be, Pb
Shutter penetration
Bulk shield
Circular entrance aperture
Collimator entrance
Drift to T0 chopper
T0 Chopper
T0 chopper shielding
Collimator pipe
Collimator shielding
Frame-overlap chopper
Overlap chopper shielding
Collimator pipe
Collimator shielding
Frame-definition chopper
Frame-definition chopper shield
Collimator pipe
Collimator shielding
Circular exit aperture
Collimator exit
Drift to Sample
Sample
Aluminum
Secondary Flight Path
Secondary flight Path shielding
Detector
Beamstop
```

Next is the list of *INDEX* pointers into the parameter block:

```
   1   18    0  125    0  126    0  127  134    0  135  136  143    0  144
 145  152    0  153    0  154    0  155  158  159  160  161  181
```

Finally, the parameter block, including the source wavelength distribution:

```
 90,-0.05,0.05,-0.05,0.05,2.5,.007525,6.8,.00444333,,8.95436-5,.0818145,16666.67,
    -1000,18,,,
 91,94,29.9,7.72,3.12312E16,615,170,.667,9.26,3.57,2.17,1.05,2.345,
   -1.69897,-1.697629,-1.695652,-1.692763,-1.688583,-1.682588,-1.674072,-1.662088,-1.645381,
   -1.622305,-1.592529,-1.559074,-1.514145,-1.464111,-1.402532,-1.338906,-1.26664,-1.18821,
   -1.107399,-1.026112,-.942809,-.858579,-.774199,-.690192,-.609081,-.531488,-.454976,
   -.379431,-.3062301,-.2391115,-.1722237,-.1068873,-.0482854,.0115282,.0651336,.1183418,
   .1670108,.214867,.2574302,.3018804,.350978,.401237,.452207,.501515,.5488,.595425,.641514,
   .686208,.730969,.774255,.821073,.867633,.914168,.960341,1.006113,1.056816,1.101922,
   1.141672,1.176525,1.221921,1.26082,1.293842,1.340518,1.384905,1.437163,1.491812,1.555346,
   1.629121,1.711947,1.80506,1.906193,2.018471,2.13695,2.245232,2.359127,2.4737,2.578686,
   2.675687,2.751545,2.809126,2.868367,2.910098,2.939216,2.959276,2.973029,2.982355,2.988662,
   2.992821,2.995572,2.997316,2.998524,2.999262,2.999732,3,,,
 20,5.65487E-5,707.355,827.355,14839.31,40,16666.67,,,
 20,9.42478E-5,117.0279,7257.36,12280.94,20,16666.67,,,
 20,9.42478E-5,82.747,16253.48,22874.19,20,16666.67,,,,
 30,.0337991,0.01,2,50,,
 43,23,1.89712,34370,49665.1,37,0.01,0.1,,16666.67,-0.32,0.32,128,0.005,.0034,
    -.3231944,.3168057,128,0.005,.0034,23/
```

The program MC_RUN asks the user for the geometry file name, the number of histories to be detected, and a starting value for the random-number generator. The user chooses whether the

recorded monitor spectrum is to represent a sample-in or a sample-out measurement. Options are also given to convert the final output histograms, which are sums of weights, to integers. There are two ways to do this. If one wishes the error bars on the output to be similar to a Poisson distribution so they look like real data, and if the error from the number of histories is small, then the value for each bin can 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. The following file is the output using the above geometry file.

```
 MONTE CARLO SIMULATION OF A NEUTRON SCATTERING INSTRUMENT
 MC_RUN code version: PC, August 9, 1995, P.A.Seeger
```

(At this point the text from the front of the geometry file is reproduced)

```
 Successfully read input file with  25 surfaces,  28 regions,
                                and 181 element parameters.
```

A test particle is sent along the instrument axis as a simple sanity check:

```
 Beam elements along the instrument axis:
 Surf  1, Z =   .000 m, enter reg  3, Shutter penetration
 Surf  3, Z =  2.498 m, enter reg  5, Circular entrance aperture
 Surf  5, Z =  2.500 m, enter reg  7, Drift to T0 chopper
 Surf  6, Z =  2.600 m, enter reg  8, T0 Chopper
 Surf  8, Z =  2.900 m, enter reg 10, Collimator pipe
 Surf 10, Z =  3.100 m, enter reg 12, Frame-overlap chopper
 Surf 12, Z =  3.102 m, enter reg 14, Collimator pipe
 Surf 13, Z =  6.000 m, enter reg 16, Frame-definition chopper
 Surf 15, Z =  6.002 m, enter reg 18, Collimator pipe
 Surf 16, Z =  6.800 m, enter reg 20, Circular exit aperture
 Surf 18, Z =  6.802 m, enter reg 22, Drift to Sample
 Surf 19, Z =  7.000 m, enter reg 23, Sample
 Surf 20, Z =  7.001 m, enter reg 24, Aluminum
 Surf 21, Z =  7.013 m, enter reg 25, Secondary Flight Path
 Surf 24, Z = 12.950 m, enter reg 28, Beamstop
 Surf 23, Z = 13.000 m, enter reg 27, Detector
 ** No exit found from Detector
```

Having found the source, the sample, and the detector, it is safe to proceed:

```
 Number of detector bins = 128 x 128 ( 90 radial), and   37 spectrum slices
 Output data forms: T       RT
 Initial random number = 075BCD15,  300000 neutrons to be detected
 Moderator phase space to be sampled =   .00060 mm**2-ster
                                      x 6.82 e-fold of energy
 Source brightness = 3.123E+16 n/ster/m**2/MW/s
```

A tally is kept of how much neutron statistical weight was absorbed in each element:

```
t     7m, 13m, 15A, 60Hz (1000us), Q=0.01/A   MC__950908_104219
Summary of neutron losses by region:
11.59% lost in region  3, Shutter penetration
     .02% lost in region  6, Collimator entrance
   13.35% lost in region  8, T0 Chopper
   73.70% lost in region 12, Frame-overlap chopper
     .71% lost in region 16, Frame-definition chopper
     .00% lost in region 20, Circular exit aperture
     .00% lost in region 23, Sample
     .07% lost in region 24, Aluminum
     .00% lost in region 25, Secondary Flight Path
```

```
    .00% lost in region 27, Detector
    .00% lost in region 28, Beamstop

Beam is centered at (   -.03,  -3.07) mm, with rms (  7.7,   7.7) mm
Total histories tracked:  4913818 (3.178E+08 neutrons)
Detected in Transmission:  303746 (1.040E+06,    .33%)
Detected in Scatter Mode:  300000 (5.302E+05,    .17%)
Beam power on target:        17.0      Mw-s
Bad-Frame neutrons passed:          1.023E+00
Bad-Frame neutrons detected:        0.000E+00
Final random number: 31BD8950
```

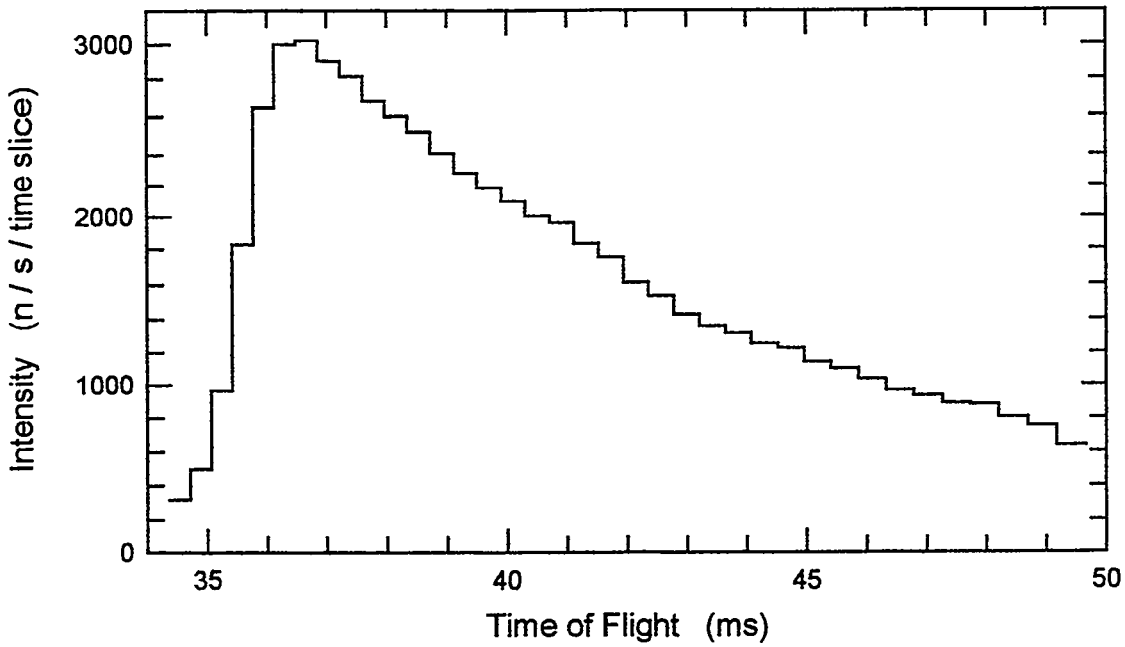

Figure 5. Time-of-flight spectrum of transmitted neutrons for LPSS test case.
At 13 m flight path length, the corresponding wavelength range is 10.5–15.1 Å. The source is a 1-MW spallation source with a coupled liquid H_2 moderator, pulsed at 60 Hz.

The output data files include header information and blocks of ASCII data for bin boundaries, data, and standard deviations. The values in the blocks are "," separated and "/" terminated as in the geometry file shown above. Data include the transmitted spectrum histogram (fig. 5) and a 2-dimensional histogram of counts integrated in radial zones on the detector for each time slice (fig. 6). By converting the radial data to Q for each individual wavelength (time slice) before combining, the time-of-flight wavelength resolution is maintained. The final result for this test case is shown in fig. 7.

8. Future Directions

It is the intention of the author that MCLIB remain in the public domain (see copyright notice in Appendix B). The most recent versions of the library and auxiliary codes as described above will continue to be available to any interested users from the author or from the Los Alamos Neutron Science Center (LANSCE) *via* Internet file transfer. Input of ideas, algorithms, and/or(public) code modules to expand the library is requested and encouraged. Element types 70 through 79 are reserved for individual non-standard use; please contact the author for assignment of permanent type numbers.

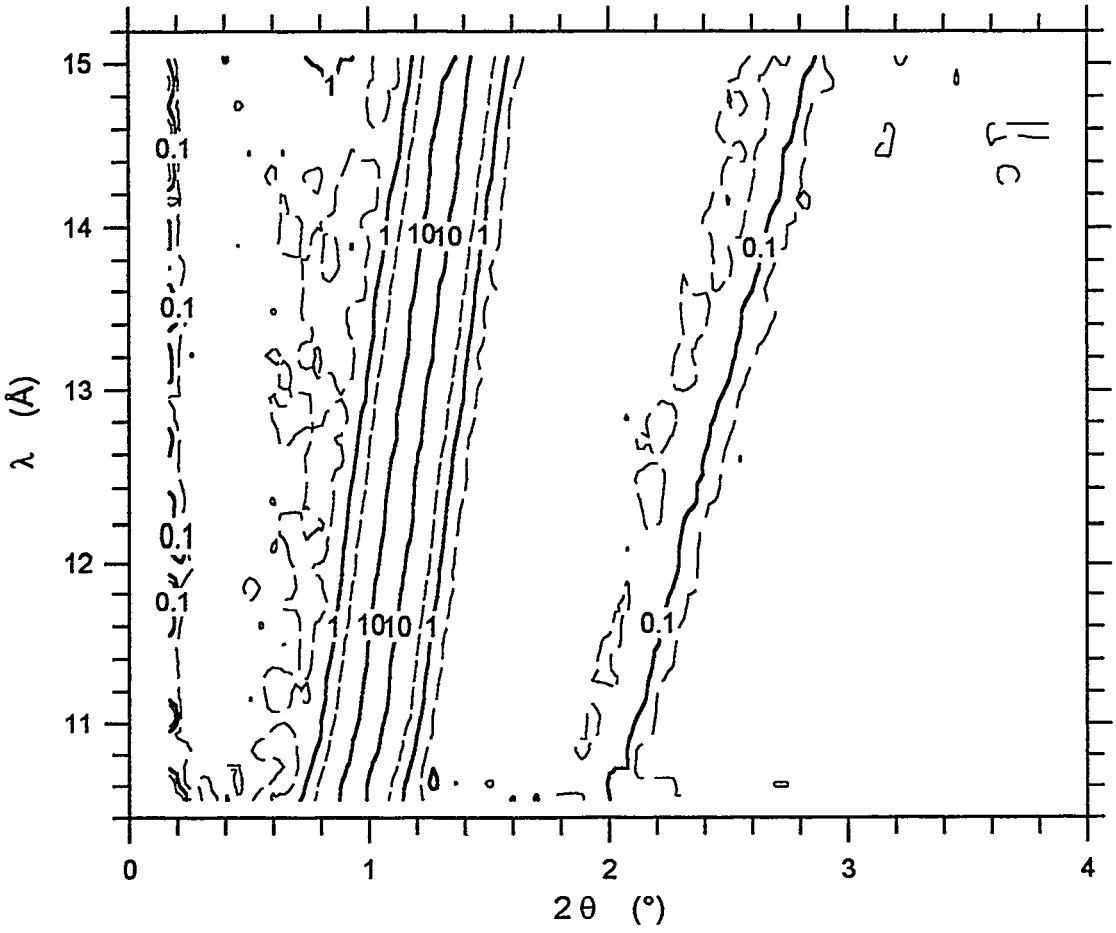

Figure 6. Relative count rate in detector *vs.* radial position and time of flight, for test case.
Time of flight has been converted to wavelength and detector radial zone to scattering angle (2θ). Contour intervals are logarithmic.

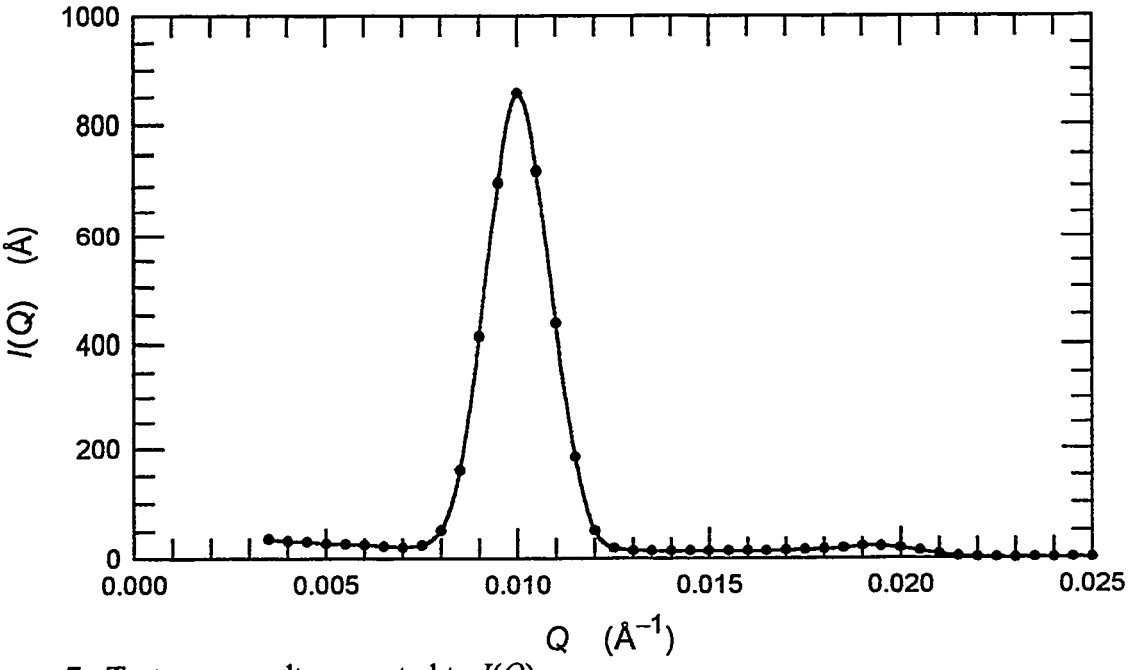

Figure 7. Test case result converted to $I(Q)$.
The scattering "sample" was a δ–function in Q at $Q = 0.010$ Å^{-1}. The standard deviation of the result is thus the resolution of the instrument at this value of Q; in this case $\sigma = 0.0009$ Å^{-1}.

A parallel (but separate) effort at LANSCE is developing a user interface (based on the Smalltalk language) which will greatly simplify design of new instrument configurations by facilitating creation of the geometry file [8]. The MCLIB library will be modified as necessary to support this project, and to include new features, such as polarization.

9. Acknowledgments

This effort has been supported historically by the U. S. Department of Energy under contract W-7405-ENG-36, and the author is grateful to the Los Alamos Neutron Science Center for their continued interest and support. Many (hopefully *most*) of the individuals who have contributed directly are listed in the text.

10. References

[1] M. W. Johnson and C. Stephanou, "MCLIB: A Library of Monte Carlo Subroutines for Neutron Scattering Problems," Rutherford Laboratory report RL-78-090, September, 1978; M. W. Johnson, "MCGUIDE: A Thermal Neutron Guide Simulation Program," Rutherford and Appleton Laboratories report RL-80-065, December, 1980.

[2] P. A. Seeger and R. Pynn, "Resolution of Pulsed-Source Small-Angle Neutron Scattering," *Nucl. Inst. Methods* **A245** (1985) 115–124.

[3] P. A. Seeger, "Monte Carlo Calculations for the Optimisation of the Beam Optics of the LOQ Spectrometer," report to Rutherford Appleton Laboratory, 28 April, 1994.

[4] P. A. Seeger, G. S. Smith, M. Fitzsimmons, G. A. Olah, L. L. Daemen, and R. P. Hjelm, Jr., various contributions, Instrumentation for a Long-Pulse Spallation Source Workshop, Lawrence Berkeley Laboratory, Berkeley, CA, April 18–21, 1995.

[5] G. J. Russell, E. J. Pitcher, and P. D. Fergusson, "The Calculated Performance of a Benchmark Target/Moderator System for an LPSS," Instrumentation for a Long-Pulse Spallation Source Workshop, Lawrence Berkeley Laboratory, Berkeley, CA, April 18–21, 1995.

[6] S. Ikeda and J. M. Carpenter, "Wide-energy-range, high-resolution measurements of neutron pulse shapes of polyethylene moderators," *Nucl. Inst. Methods* **A239** (1985) 536–544.

[7] R. A. Robinson and J. M. Carpenter, "On the use of switch functions in describing pulsed neutron moderators," *Nucl. Inst. Methods* **A307** (1991) 359–365.

[8] T. G. Thelliez, L. L. Daemen, P. A. Seeger, and R. P. Hjelm, Jr., (in preparation), Proceedings of the 13th Meeting of the International Collaboration on Advanced Neutron Sources, October 11–14, 1995, Paul Scherrer Institut, Villigen, Switzerland.

Appendix A

Files MC_GEOM.INC, MC_ELMNT.INC and CONSTANT.INC

MC_GEOM.INC

```
C  08 Feb 1994: converted from COMMON to STRUCTUREs.  Added WT, M, and POL to
C               particle record.  Require Y direction to be vertical (for
C               gravity). [PAS]
C  17 Feb 1994: added BETA to surface record; type 5 boundary; made MAXP
C               a parameter [PAS]
C  15 Feb 1995: include structure /MC_ELEMENT/ formerly in MC_ELMNT.INC [PAS]
C
C  To change dimensions of all arrays, change maximum numbers of
C    surfaces (MAXS), regions (MAXR), and/or element parameters (MAXP) in
C    following PARAMETER statement.
      INTEGER*4 MAXS, MAXR, MAXP
      PARAMETER (MAXS=150, MAXR=100, MAXP=800)
C
C  Definitions of surface parameters and array of boundaries of surfaces:
C    SURFACE is a record containing 10 coefficients of a general
C      quadratic surface, of the form
C      A*(X**2) + B*X + C*(Y**2) + D*Y + E*(Z**2) + F*Z + G +
C          P*(X*Y) + Q*(Y*Z) + R*(Z*X)  =  0
C      BETA = surface roughness parameter between 0 (smooth) and 1 (cosine);
C             negative or >1 is completely random
      STRUCTURE /SURFACE/
         REAL*4 A, B, C, D, E, F, G, P, Q, R
         REAL*4 BETA
      END STRUCTURE
C
C    REGION is record containing NSURF values of the I*2 variable IGEOM,
C      defining a region by its bounding surfaces:
C      +  if interior of region is on + side of surface
C      -  if interior of region is on - side of surface
C      0  if surface is not a boundary of the region
C      1  for ordinary surface described by roughness BETA and possibility
C         of refraction or critical reflection
C      2  for totally reflecting surface
C      3  for diffuse scattering surface
C      4  for absorbing surface
C      5  special action required in previous region before crossing surface
      STRUCTURE /REGION/
         INTEGER*2 IGEOM(MAXS)
      END STRUCTURE
C
C    NSURF = number of surfaces defined
C    NREG = number of regions defined
      STRUCTURE /MC_GEOM/
         INTEGER*4 NSURF,NREG
         RECORD /SURFACE/ SURFACE(MAXS)
         RECORD /REGION/ REGION(MAXR)
      END STRUCTURE
C
C  Definitions of position and velocity of particle:
C    (X,Y,Z) = position of particle (m); note that +Y is UP (for gravity)
C    (VX,VY,VZ) = velocity of particle (m/us)
C    TOF = time of flight of particle (us)
C    M = atomic number of particle (e.g., 1 for neutron, 0 for photon)
C    POL = polarization of particle, in range -1.0 to +1.0; 0 for unpolarized
```

```
C    WT = statistical weight of particle
      STRUCTURE /PARTICLE/
         REAL*4 X, Y, Z, VX, VY, VZ, TOF, M, POL, WT
      END STRUCTURE
C
C  Definitions of element-parameter block
C  Each INDEX points to the beginning of a structure within the contiguous
C    block PARAM. The first entry in PARAM for each element identifies the
C    type, followed by a varying number of parameters.
C    NAME = 40-character descriptive name of a region or element
C    NEXTINDX = pointer to next available location in PARAM
C    INDEX = pointer into parameter block; if 0, region is not an "element"
C    PARAM = block of element parameters
      STRUCTURE /MC_ELEMENT/
         CHARACTER*40 NAME(MAXR)
         INTEGER*4  NEXTINDX, INDEX(MAXR)
         REAL*4  PARAM(MAXP)
      END STRUCTURE
C
```

MC_ELMNT.INC

```
C  Definitions of beam elements which may occur in regions, and their parameters
C
C  P. A. Seeger, April 20, 1994
C    04 Jan 1995: define offsets of parameters within blocks, rather than
C                 structure with UNIONs [PAS]
C    10 Jan 1995: modified source type 90; added type 91 for source spectrum
C                 and lineshape description [PAS]
C    01 Feb 1995: added types 5 (Be), 13 (crystal monochromator), and 35
C                 (reflectometry) [PAS]
C    15 Feb 1995: modified type 91; moved structure definition to MC_GEOM.INC
C                 [PAS]
C    04 Mar 1995: types 90.n for rectangular and/or offset phase space [PAS]
C    09 Mar 1995: type 44 (longitudinal detector); all detectors need surface
C                 number [PAS]
C    06 Jun 1995: 2 more parameters in monochromator type 13 [PAS]
C    13 Jul 1995: type 36, general powder [PAS,Uli Wildgruber,Luke Daemon]
C    04 Aug 1995: type 6, single-crystal filter [PAS]
C    26 Aug 1995: type 32, isotropic scatterer [PAS]
C    05 Sep 1995: revised parameters for pulse shape (type=91) [PAS]
C
C  The first entry in PARAM for each element identifies the
C    type, followed by a varying number of parameters.
C
      INTEGER ELMNT_TYPE
      PARAMETER (ELMNT_TYPE=0)
C  type 0 = total absorber; no additional parameters
C  type 1 = amorphous unpolarized material; 4 parameters
C       Real and Imaginary scattering-length density (10**10/cm**2)
C       macroscopic scattering cross section (1/m)
C       velocity-dependent cross section, at 1 m/us (1/us)
        INTEGER REAL_RHO, IMAG_RHO, NSIGMA0, NSIGMAV
        PARAMETER (REAL_RHO=1,IMAG_RHO=2,NSIGMA0=3,NSIGMAV=4)
C  type 2 = aluminum, including Bragg edges; no additional parameters
C  type 3 = hydrogenous, including multiple scattering; 1 parameter
C       Relative hydrogen density compared to water
        INTEGER H_DENSITY
        PARAMETER (H_DENSITY=1)
```

```
C  type 4 = supermirror characterized by 2 scattering densities; 5 parameters
C        Complex bulk material scattering-length density (10**10/cm**2)
C        Complex effective supermirror density (10**10/cm**2)
C        Relative efficiency of supermirror regime
         INTEGER REAL_BULK, IMAG_BULK, REAL_SUPER, IMAG_SUPER, EFF_SUPER
         PARAMETER (REAL_BULK=1,IMAG_BULK=2,REAL_SUPER=3,IMAG_SUPER=4,
     1             EFF_SUPER=5)
C  type 5 = beryllium at 100K, including Bragg edges; no additional parameters
C  type 6 = single-crystal filter, Freund formalism; 3 parameters
C           xsec = sigfree*(1-exp(-C2/lambda**2)) + sigabs*lambda
C        Limiting (short wavelength) free-atom macroscopic cross section (1/cm)
C        -ln(1 - (sig(1A)-sigabs)/(sigfree-sigabs)) (A**2)
C        sum of 1/v macroscopic cross sections at 1A (1/cm/A)
         INTEGER XSIGFREE, X_C2, XSIGABS
         PARAMETER (XSIGFREE=1,X_C2=2,XSIGABS=3)
C
C  type 10 = multi-aperture collimator
C  type 11 = multi-slit collimator, vertical blades; 3 or 5 parameters
C        Spacing of slits, centerline-to-centerline (m)
C        Rate of convergence (>0) or divergence (<0) of one slit
C        Z at entrance of the region, where spacing is measured (m)
         INTEGER C_DELTA, C_TAPER, C_ZENTER
         PARAMETER (C_DELTA=1,C_TAPER=2,C_ZENTER=3)
C        For a curved system (bender),
C           sine of half the angle of bend
C           cosine of half the angle of bend
         INTEGER B_SIN_PHI, B_COS_PHI
         PARAMETER (B_SIN_PHI=4,B_COS_PHI=5)
C  type 12 = multi-slit collimator, horizontal blades; 5 parameters
C           Same parameters as type 11
C  type 13 = crystal monochrometer; 10 parameters
C        Twice the crystal plane spacing (A)
C        Nominal Z position for rotation of instrument axis (m)
C        Sine and cosine of take-off angle
C        X-, Y-, and Z-components of mosaic spread, rms of sines of angles
C        rms spread of plane spacing, (delta d)/d
C        max number of loops (or microcrystal orientations) to try
C        probability normalization factor per try, derived from reflection
C          probability at peak wavelength: 1 - (1-max_prob)**(1/trys)
         INTEGER M_2D_SPACE, M_Z0, M_SIN_2TH, M_COS_2TH, M_ROTX,
     1           M_ROTY, M_ROTZ, M_D_SPREAD, M_TRY, M_PROB
         PARAMETER (M_2D_SPACE=1,M_Z0=2,M_SIN_2TH=3,M_COS_2TH=4,
     1             M_ROTX=5,M_ROTY=6,M_ROTZ=7,M_D_SPREAD=8,
     2             M_TRY=9,M_PROB=10)
C
C  types 20.n = chopper (disk or blade); 6 parameters
C          .0 or .2 for motion in x-direction, .1 or .3 for vertical
C          .2 or .3 is counter-rotating (fully closed when edges at 0)
C        Linear velocity of opening crossing beam centerline (m/us)
C        Time to cover or uncover half the width of the moderator (us)
C        Nominal time at which opening chopper edge crosses zero (us)
C        Nominal time at which closing chopper edge crosses zero (us)
C        Phase jitter of chopper, rms (us)
C        Period of chopper (us)
         INTEGER CHP_VEL, CHP_HALF, CHP_OPEN, CHP_CLOSE,
     1           CHP_JITTER, CHP_PERIOD
         PARAMETER (CHP_VEL=1,CHP_HALF=2,CHP_OPEN=3,CHP_CLOSE=4,
     1             CHP_JITTER=5,CHP_PERIOD=6)
```

```
C  type 21 = Fermi chopper
C  type 22 = gravity focuser; 5 parameters
C       acceleration (m/us**2), and rms phase jitter (us)
C       nominal times for start and top of upward stroke (us)
C       time between pulses (us)
        INTEGER G_ACCEL, G_JITTER, G_START, G_TOP, G_PERIOD
        PARAMETER (G_ACCEL=1,G_JITTER=2,G_START=3,G_TOP=4,G_PERIOD=5)
C  type 23 = removable beamstop; no additional parameters
C
C       first parameter of most samples is -ln(transmission at 1 A)
        INTEGER SIGMA_1A
        PARAMETER (SIGMA_1A=1)
C  type 30 = sample which scatters at constant Q; 1 additional parameter
C          value of Q for scatter (1/A)
           INTEGER Q_SCATTER
           PARAMETER (Q_SCATTER=2)
C  type 31 = scattering sample of hard spheres; 1 additional parameter
C          hard-sphere radius for scatter (A)
           INTEGER R_SPHERE
           PARAMETER (R_SPHERE=2)
C  type 32 = isotropic scatterer with constant energy change; 1 additional
C          inelastic energy change (0 if elastic) (meV)
           INTEGER DELTA_E
           PARAMETER (DELTA_E=2)
C  type 34 = inelastic scattering using MCNP file; no parameters, but NAME
C       must be '[path]filename' for the file
C  type 35 = scattering from layered reflectometry sample; 1 + 4*N parameters
C       number of layers, including substrate
C       parameters for each layer, starting with substrate:
C          4pi*Real and Imaginary scattering-length density (1/A**2)
C          Thickness of layer (zero for substrate) (A)
C          Roughness, 2*sigma**2 of outer surface of this layer (A**2)
        INTEGER NLAYERS, REAL4PINB, IMAG4PINB, THK_LAYER, ROUGHNESS
        PARAMETER (NLAYERS=1, REAL4PINB=2, IMAG4PINB=3,
     1             THK_LAYER=4, ROUGHNESS=5)
C  type 36 = scattering from isotropic polycrystalline powder; 6+2*N parameters
C       number of Bragg edges included
C       limiting (short wavelength) macroscopic total xsection (1/cm)
C       macroscopic incoherent scattering xsection (1/cm)
C       macroscopic 1/v scattering xsection at 1 A (1/cm/A)
C       macroscopic 1/v absorption xsection at 1 A (1/cm/A)
C       table of d-spacings of Bragg edges (A), followed by explicit 0 and
C         table of cumulative macroscopic xsections at 1 A (1/cm/A**2)
        INTEGER N_BRAGG, PSIGMAT, PSIGMAI, PSIGMAS, PSIGMAA, D_BRAGG
        PARAMETER (N_BRAGG=1,PSIGMAT=2,PSIGMAI=3,PSIGMAS=4,PSIGMAA=5,
     1             D_BRAGG=6)
C
C  type 40 = detector; 9 additional parameters
C       first parameter of all detectors is surface number
C       second parameter of all detectors is -ln(1 - efficiency at 1 A)
        INTEGER DET_SURF, D_ALPHA_1A
        PARAMETER (DET_SURF=1,D_ALPHA_1A=2)
C       time-of-flight clock parameters:
C         minimum and maximum times (us)
C         number of time channels
C         if logarithmic, dt/t (otherwise dt/t = 0)
C         minimum clock tick in determining log scale (us)
```

```
C          repeat period of data-acquisition electronics (us)
C        Note: if t-o-f is logarithmic, TMAX is overridden
         INTEGER D_TMIN, D_TMAX, D_TCHANS, D_DT_OVER_T, D_TICK, D_DELAY,
     1           D_T_PERIOD
         PARAMETER (D_TMIN=3,D_TMAX=4,D_TCHANS=5,D_DT_OVER_T=6,
     1              D_TICK=7,D_DELAY=8,D_T_PERIOD=9)
C  type 41 = vertical linear detector; 14 additional parameters
C           locations of bottom and top of detector (m)
C           number of detector elements
C           size of detector element (m)
C           root-mean-square encoding error of detector (m)
            INTEGER DET_YMIN, DET_YMAX, DET_NY, DET_DELY, DET_RMSY
            PARAMETER (DET_YMIN=10,DET_YMAX=11,DET_NY=12,DET_DELY=13,
     1                 DET_RMSY=14)
C  type 42 = horizontal linear detector; 14 additional parameters
C           locations of left and right ends of detector (m)
C           number of detector elements
C           size of detector element (m)
C           root-mean-square encoding error of detector (m)
            INTEGER DET_XMIN, DET_XMAX, DET_NX, DET_DELX, DET_RMSX
            PARAMETER (DET_XMIN=10,DET_XMAX=11,DET_NX=12,DET_DELX=13,
     1                 DET_RMSX=14)
C  type 43 = 2-D rectilinear detector; 18 additional parameters
C           locations of left and right edges of detector (m)
C           number of detector elements in the horizontal direction
C           width of detector element (m)
C           root-mean-square X encoding error of detector (m)
C           locations of bottom and top edges of detector (m)
C           number of detector elements in the vertical direction
C           height of detector element (m)
C           root-mean-square Y encoding error of detector (m)
            INTEGER DET2_XMIN, DET2_XMAX, DET2_NX, DET2_DELX, DET2_RMSX
            INTEGER DET2_YMIN, DET2_YMAX, DET2_NY, DET2_DELY, DET2_RMSY
            PARAMETER (DET2_XMIN=10,DET2_XMAX=11,DET2_NX=12,
     1                 DET2_DELX=13,DET2_RMSX=14,DET2_YMIN=15,
     2                 DET2_YMAX=16,DET2_NY=17,DET2_DELY=18,
     3                 DET2_RMSY=19)
C  type 44 = longitudinal linear detector; 14 additional parameters
C           locations of upstream and downstream ends of detector (m)
C           number of detector elements
C           size of detector element (m)
C           root-mean-square encoding error of detector (m)
            INTEGER DET_ZMIN, DET_ZMAX, DET_NZ, DET_DELZ, DET_RMSZ
            PARAMETER (DET_ZMIN=10,DET_ZMAX=11,DET_NZ=12,DET_DELZ=13,
     1                 DET_RMSZ=14)
C
C  type 50 = scattering chamber, void-filled.  No parameters, but other regions
C        may be embedded, indicated by surface types with 10s digit on.
C
C  types 90.n = source size and phase space to be sampled; 14-18 parameters
C        edges of rectangular moderator face (m)
         INTEGER MOD_XMIN, MOD_XMAX, MOD_YMIN, MOD_YMAX
         PARAMETER (MOD_XMIN=1,MOD_XMAX=2,MOD_YMIN=3,MOD_YMAX=4)
C        location and radius (half-width) of apertures which define beam (m)
         INTEGER APTR1_Z, APTR1_R, APTR2_Z, APTR2_R
         PARAMETER (APTR1_Z=5,APTR1_R=6,APTR2_Z=7,APTR2_R=8)
C        additional vertical space to sample for gravity focus (m)
         INTEGER G_DELY2
```

```
      PARAMETER (G_DELY2=9)
C     min and max neutron energy to be sampled (eV)
C     time between beam pulses and square pulse width (us)
      INTEGER S_EMIN, S_EMAX, S_PERIOD, S_WIDTH
      PARAMETER (S_EMIN=10,S_EMAX=11,S_PERIOD=12,S_WIDTH=13)
C     offset to parameter block with spectrum and lineshape parameters
      INTEGER E_OFFSET
      PARAMETER (E_OFFSET=14)
C     (optional) half-heights of apertures, type 90.1 or 90.2 (m)
      INTEGER APTR1_Y, APTR2_Y
      PARAMETER (APTR1_Y=15,APTR2_Y=16)
C     (optional) vertical offsets of apertures, type 90.4 (m)
      INTEGER APTR1_Y_OFFSET, APTR2_Y_OFFSET
      PARAMETER (APTR1_Y_OFFSET=17,APTR2_Y_OFFSET=18)
C  type 91 = source energy distribution table and lineshape parameters; 12
C     parameters plus length of table
C     number of entries in energy table (1 is special case)
C     location in table of center of normal distribution (index units)
C       or value of nominal neutron velocity (m/us) (if # entries = 1)
C     standard deviation of normal distribution (table index units)
C       or relative fwhm of velocity selector (if # entries = 1)
      INTEGER N_E_TABLE, CENT_TABLE, SIGMA_TABLE
      PARAMETER (N_E_TABLE=1,CENT_TABLE=2,SIGMA_TABLE=3)
C     source brightness, summed over limits of E_TABLE (n/ster/m**2/MW/s)
      INTEGER S_BRIGHT
      PARAMETER (S_BRIGHT=4)
C     1 or 2 exponential time constants in thermal (low-energy) limit (us)
C     probability of 2nd exponential
      INTEGER TAU_TH1, TAU_TH2, TAU2_RATIO
      PARAMETER (TAU_TH1=5,TAU_TH2=6,TAU2_RATIO=7)
C     epithermal (high energy) time constant proportional to lambda (us/A)
C     Gaussian delay and width proportional to lambda (us/A)
      INTEGER TAU_EPI, T_DELAY, T_WIDTH
      PARAMETER (TAU_EPI=8,T_DELAY=9,T_WIDTH=10)
C     switching function 1/e point (A) and power (slope)
      INTEGER SWITCH_LAMBDA, SWITCH_POWER
      PARAMETER (SWITCH_LAMBDA=11,SWITCH_POWER=12)
C     origin of table of cumulative energy distribution (weighted by
C        lambda**2) of source spectrum on equally spaced normal-curve values
C        of log(energy/1meV)
      INTEGER E_TABLE
      PARAMETER (E_TABLE=13)
```

CONSTANT.INC

```
C  Various physical constants
C
C    GOVER2 = half the acceleration of gravity (m/us**2)
C    HOVERM = Plank's constant/neutron mass (m-A/us)
C    HSQOV2M = Plank's constant squared over 2 Mneutron (eV-A**2)
C    ROOTM_2 = square root of half the neutron  mass (eV**0.5-us/m)
C    TWOPI = 2*pi
      REAL*4 GOVER2, HOVERM, HSQOV2M, ROOTM_2, TWOPI
      PARAMETER (GOVER2   = 4.858E-12,
     1           HOVERM   = 0.0039560339,
     2           HSQOV2M  = 0.0818145347,
     3           ROOTM_2  = 72.298,
     4           TWOPI    = 6.283185307)
```

Appendix B
Subroutine Abstracts

```
C
C         Copyright, 1995,  The  Regents  of the University of California.
C     This software was produced under a U.S. Government contract (W-7405-
C     ENG-36)  by the Los Alamos  National  Laboratory,  which is operated
C     by the University of  California for the U. S. Department of Energy.
C     The U. S. Government  is licensed to use,  reproduce, and distribute
C     this software.  Permission is granted  to the public to copy and use
C     this  software  without charge,  provided  that  this notice and any
C     statement  of authorship  are reproduced on  all copies. Neither the
C     Government nor the University makes any warranty, express or implied,
C     or assumes any liability or responsibility for the use of this soft-
C     ware.
C

C++
C********** A N G L I **********
C
      SUBROUTINE ANGLI(PART, SURF, COSTH, AP, BP, CP)
C
C  Compute the cosine of the angle of incidence of the particle PART
C    with respect to surface SURF.  The cosine is returned in COSTH,
C    and the direction cosines of the normal to the surface as AP,BP,CP.
C    No test is made to assure that particle is actually on the surface.
C
C  From subroutine WOBBLE, M. W. Johnson, Rutherford and Appleton Laboratories
C    report RL-80-065; P. A. Seeger, 1984.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; changed calling
C                 sequence  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record containing particle coordinates (input)
C    SURF = record containing definition of surface (input)
C    COSTH = cosine of angle between /MC_PART/ and surface normal (output)
C    AP,BP,CP = direction cosines of surface normal (output)
      RECORD /PARTICLE/ PART
      RECORD /SURFACE/ SURF
      REAL*4 COSTH, AP, BP, CP
C
C  No externals
C--

C++
C********** A T T E N _ A L **********
C
      REAL*4 FUNCTION ATTEN_AL(LAMBDA)
C
C   Neutron attenuation length (mm) of polycrystalline aluminum at 300K,
C    as a function of neutron wavelength LAMBDA in Angstroms.
C
```

```
C  From J. M. Carpenter et al., SIGAL, January 16, 1986 (IPNS Note 32)
C  28 Jan 1993:  adapted, restructured, modified numeric values [PAS]
C
C  Variables in calling sequence:
C    LAMBDA = neutron wavelength (A) (input)
      IMPLICIT NONE
      REAL*4 LAMBDA
C
C  No externals
C--

C++
C********** A T T E N _ B E **********
C
      REAL*4 FUNCTION ATTEN_BE(LAMBDA)
C
C   Neutron attenuation length (mm) of polycrystalline beryllium at 100K,
C    as a function of neutron wavelength LAMBDA in Angstroms.
C
C  From J. M. Carpenter et al., SIGAL, January 16, 1986 (IPNS Note 32)
C  25 Jan 1995: converted from Al to Be [PAS & Bob VonDreele]
C
C  Variables in calling sequence:
C    LAMBDA = neutron wavelength (A) (input)
      IMPLICIT NONE
      REAL*4 LAMBDA
C
C  No externals
C--

C++
C********** A T T E N _ X **********
C
      REAL*4 FUNCTION ATTEN_X(PARAMS, LAMBDA)
C
C   Neutron attenuation length (mm) of a single-crystal filter, using the
C     formalism of Freund, as a function of neutron wavelength LAMBDA:
C     sigma = sigfree*[1-exp(-C2/lambda**2)] + sigabs*lambda
C
C  From A. K. Freund, Nucl. Inst. Methods 213 (1983) 495-501, eq. 12.
C  P. A. Seeger, August 4, 1995.
C    14 Aug 1995: was returning reciprocal [PAS,GXu]
C
C  Variables in calling sequence:
C    PARAMS = array with cross section parameters (input)
C    LAMBDA = neutron wavelength (A) (input)
      IMPLICIT NONE
      INCLUDE  'mc_elmnt.inc'
      REAL*4   PARAMS(*), LAMBDA
C
C  No externals
C--

C++
C********* D I G I T S **********
C
      SUBROUTINE DIGITS(Q,STRING,N)
```

```
C  Format a floating-point number into 10 or fewer ASCII characters.
C    This subroutine encodes a floating-point number Q into a character string
C    no longer than STRING (maximum 10), with no embedded blanks or trailing
C    insignificant zeros.  The actual length is returned as N.  Underflows are
C    returned as '.0000' and overflows as '*****'.  Zero Q becomes '0'.
C
C  P. A. Seeger, Los Alamos National Laboratory, May 24, 1986
C    05 Dec 1987: improved elimination of trailing insignificant digits [PAS]
C
C  No Externals
C--

C++
C**********  D I S T  **********
C
      REAL*4 FUNCTION DIST(PART, SURF, ON_SURF)
C
C  Find distance along trajectory of particle PART to surface SURF.  For
C    particles with mass, gravity is included in determining intersection, but
C    the distance returned assumes constant velocity.  If logical ON_SURF is
C    .TRUE., particle is assumed to be on the surface already, but test is
C    made for second intersection with quadratic surface.  If trajectory does
C    not intersect the surface, DIST will be -10**38 on return.  No parameters
C    of the particle are changed.
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
C    Modified by P.A. Seeger (1980) to include test for being on surface ISURF.
C    Restructured by P.A. Seeger, 1984.
C    08 Feb 1994: converted from COMMON to STRUCTUREs; changed calling
C                 sequence; added gravity  [PAS]
C    11 Apr 1994: changed test of AA << BB to avoid underflow  [PAS]
C    06 Mar 1995: solve in double precision [PAS]
C    23 Mar 1995: expand square root if BB**2 >> CC/AA [MFitz,PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
      INCLUDE 'constant.inc'
C
C  Variables in calling sequence
C    PART = record with particle coordinates (input)
C    SURF = record with surface definition (input)
C    I = surface to which distance is being found (input)
C    ON_SURF = flag that particle is on the surface in question (input)
      RECORD /PARTICLE/ PART
      RECORD /SURFACE/ SURF
      LOGICAL ON_SURF
C
C  No externals.
C--

C++
C********** D T O E X **********
C
      SUBROUTINE DTOEX(PART, GEOM, IREG, ISURF, JSURF, EXDIST)
C
C  Compute distance EXDIST to nearest boundary of region IREG from
C    position and velocity given in PART, using surface and region
```

```
C    definitions in GEOM.  If IREG = 0, finds distance to nearest
C    surface in forward direction.  If ISURF is not 0, the particle
C    is assumed to be on that surface already, but may intersect it again
C    if the surface is quadratic.  On return, JSURF is the exit surface
C    and EXDIST is the distance to it; the particle has not been moved.
C    If no exit is found, JSURF will be 0 and EXDIST will be negative.
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
C    Modified by P.A. Seeger (1980) for case of particle on surface ISURF.
C    02 Jan 1985: Included JSURF as separate entry in calling sequence [PAS]
C    03 Feb 1994: converted from COMMON to STRUCTUREs; added PART and GEOM
C                 to calling sequence  [PAS]
C    06 Mar 1995: allow zero as a result [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input)
C    GEOM = structure with definitions of all surfaces and regions (input)
C    IREG = region particle is in (input)
C    ISURF = surface which the particle PART is on (input)
C    JSURF = surface the direction in PART is pointed toward (output)
C    EXDIST = distance to surface JSURF (output)
      RECORD    /PARTICLE/ PART
      RECORD    /MC_GEOM/  GEOM
      INTEGER*4 IREG, ISURF, JSURF
      REAL*4    EXDIST
C
C  Externals:
C    DIST
      REAL*4 DIST
C--

C++
C********** E L S C A T **********
C
      SUBROUTINE ELSCAT(VX, VY, VZ, STH, ISEED)
C
C  Elastic scattering from original velocity (VX,VY,VZ) at random
C    azimuthal angle about the velocity vector, at a scattering angle
C    given by  STH = 2 sin(theta/2).  If the velocity is 0 or if the
C    absolute value of STH is greater than 2, no scattering occurs.
C
C  P. A. Seeger, 1984.
C    28 Mar 1994: test VX**2 vs. V**2 instead of VX vs. V [PAS]
C    26 Aug 1995: protect against negative square root [PAS]
C
C  Variables in calling sequence:
C    VX, VY, VZ = vector velocity (input/output)
C    STH = twice the sine of half the scattering angle (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      INTEGER*4 ISEED
      REAL*4 VX,VY,VZ,STH
C
```

```
C    RNDCRCL
C--

C++
C**********   G A M M L N   **********
C
      REAL*4 FUNCTION GAMMLN(XX)
C
C  Returns natural log of gamma(XX), for XX > 0.
C
C  Press et al., "Numerical Recipes" (2nd ed., Cambridge, 1992) p. 207
C
C  Variables in calling sequence:
C    XX = argument (input)
      IMPLICIT NONE
      REAL*4   XX
C
C  No externals
C--

C++
C**********  G E T _ B I N S  **********
C
      SUBROUTINE GET_BINS(PARAMS, IXMAX, IYMAX, ITMAX,
     1                    DELX, DELY, T_BINS, XMIN, YMIN)
C
C  Look up the binning parameters for the detector given in structure
C    PARAMS, and return dimensions, pixel sizes, the time-slice boundaries
C    (note that number of boundaries is 1 more than number of bins), and
C    minimum horizontal and vertical detector edges.
C
C  P. A. Seeger, April 10, 1994
C    04 Jan 1995: MC_ELMNT.INC now has parameter offsets, not UNIONs [PAS]
C    28 Mar 1995: new calling sequence to return XMIN and YMIN; added type 44;
C                 default DELX and DELY = 0 (was 1 m) [PAS]
C    12 Apr 1995: allow linear t bins; test for t valid scale; offset t [PAS]
C    26 Jul 1995: linear scale always started at zero [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
      INCLUDE 'mc_elmnt.inc'
C
C  Variables in calling sequence:
C    PARAMS = array with description of a detector (input)
C    IXMAX = number of horizontal detector elements (output)
C    IYMAX = number of vertical detector elements (output)
C    ITMAX = number of time bins (output)
C    DELX,DELY = size of detector elements (m) (output)
C    T_BINS = array of ITMAX+1 time bin boundaries (us) (output)
C    XMIN,YMIN = detector boundaries, horizontal and vertical (m) (output)
      INTEGER IXMAX, IYMAX, ITMAX
      REAL*4  PARAMS(*), DELX, DELY, T_BINS(*), XMIN, YMIN
C
C  No Externals
C--

C++
```

```
C**********  G E T _ R H O  **********
C
      COMPLEX*8 FUNCTION GET_RHO(PARAMS, ISEED)
C
C  Look up the complex scattering-length density for a region with
C    type and parameters given in structure PARAMS.
C
C  P. A. Seeger, March 28, 1988
C    17 Apr 1994: include supermirror type (4); ISEED in calling sequence [PAS]
C    05 Jan 1995: MC_ELMNT.INC now has parameter offsets, not UNIONs [PAS]
C
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
      INCLUDE 'mc_elmnt.inc'
      REAL*4   PARAMS(*)
      INTEGER*4 ISEED
C
C  Externals:
C     RAN
      REAL*4  RAN
C--

C++
C***********    G I N I    **********
C
      LOGICAL FUNCTION GINI(X,Y,YVAR,NCHANS,INTENS,SIGINT,CENTER,
     1                      SIGCEN,SIGMA)
C
C    This routine sums data and computes the centroid and estimates the
C    standard deviation of the distribution using Gini's mean-difference
C    statistic (C. W. Akerlof, Nucl. Inst. Methods 211 (1983) 439-445).
C    The value .TRUE. is returned if successful, .FALSE. if too many
C    negative data in the array.
C
C  P. A. Seeger, Los Alamos National Laboratory, Feb. 5, 1992
C
C    Variables in calling sequence
C       X       R(*)    Input     Array of bin boundaries (NCHANS + 1)
C       Y       R(*)    Input     Array of counts-per-bin
C       YVAR    R(*)    Input     Array of variances of Y; if Y is Poisson,
C                                 this may be the same array as Y
C       NCHANS  I       Input     Length of arrays; number of bins to sum
C       INTENS  R       Output    Integrated intensity (sum of counts)
C       SIGINT  R       Output    Standard deviation of INTENS
C       CENTER  R       Output    Centroid of distribution
C       SIGCEN  R       Output    Standard deviation of CENTER
C       SIGMA   R       Output    Gini's mean-difference estimator of std. dev.
C--

C++
C**********  G R A V _ F O C  **********
C
      REAL*4 FUNCTION GRAV_FOC(T, ACCEL, JITTER, START, TOP, PERIOD,
     1                         ISEED)
C
C  Determine vertical position of gravity focusing device.
C
```

```
C
C  Variables in calling sequence:
C    T = time (measured from proton pulse) at which to determine position (us)
C    ACCEL = acceleration of gravity focuser (m/us/us)
C    JITTER = rms phase uncertainty of gravity focuser (us)
C    START = earliest time that gravity focuser must accept (us)
C    TOP = latest time, top of trajectory (us)
C    PERIOD = time between successive pulses of the gravity focuser (us)
C    ISEED = random number generator seed
      REAL*4  T, ACCEL, JITTER, START, TOP, PERIOD
      INTEGER ISEED
C
C  Externals:
C    PLNORM
      REAL*4 PLNORM
C--

C++
C**********    H U N T    **********
C
      SUBROUTINE HUNT(XX,N,X,JLO)
C
C  Find index JLO in array XX of length N, such that X is between XX(JLO)
C    and XX(JLO+1).  If JLO = 0 at entry, use full bisection; otherwise
C    begin search at initial value of JLO.
C
C  Press et al., "Numerical Recipes" (Cambridge, 1986) p. 91;
C    adapted by P. A. Seeger, April 13, 1987.
C
C  No externals
C
C  Variables in calling sequence:
C     XX      R(*)    Input     Monotonic array
C     N       I       Input     Length of array XX
C     X       R       Input     Value of which to find location in XX
C     JLO     I       In/Out    Index in XX such that XX(JLO) and XX(JLO+1)
C                               bracket X
C--

C++
C********** I  N  T  E  R  P **********
C
      SUBROUTINE INTERP(XX, X, NX, I, R)
C
C  Subroutine to interpolate a value XX in a table X of length NX.  Result
C    is index number I of the cell containing XX, and the ratio R of XX
C    into the cell.  Input array X must be monotonic increasing, with no
C    repeated values.  If XX is outside the range of the table, I will be
C    1 or NX and R will be negative or greater than one, indicating
C    extrapolation.
C
C  P. A. Seeger, 1984
C
C  Variables in calling sequence:
C    XX = value to be found in table
C    X = array in which to interpolate
C    NX = length of array X
C    I = index of table value next lower (or =) XX
```

```
C    R = ratio of location of XX to X(I+1) - X(I)
      IMPLICIT NONE
      INTEGER*4 NX, I
      REAL*4 XX, X(NX), R
C
C  No externals
C--

C++
C**********   K E R N E L   **********
C
      SUBROUTINE KERNEL(LAW, E, EPRIME, MU, ATTEN, ISEED)
c
c   Subroutine to scatter neutrons according to the scattering law
c     contained in the file named LAW, which has the format of a Type 1
c     thermal data table (NTY=4) from the MCNP library.
c
c   L.L. Daemen, P.A. Seeger - April 1, 1995
c     03 Apr 1995: eliminate local arrays, use XSS directly [PAS,LLD]
c
c   Note: local array XSS must be sized adequately to hold largest LAW file
      IMPLICIT  NONE
      INTEGER   NMAX
      PARAMETER (NMAX=18432)
      REAL*4    XSS(NMAX)
c
c   Variables in calling sequence:
c     LAW    = complete path to the file containing the scattering law (input)
c     E      = energy of the incident neutron in eV (input)
c     EPRIME = energy of the scattered neutron in eV (output)
c     MU     = cosine of the scattering angle, 2*theta (output)
c     ATTEN  = scattering mean free path in mm (output)
c     ISEED  = seed for the random number generator (input/output)
      CHARACTER LAW*40
      REAL      E, EPRIME, MU, ATTEN
      INTEGER*4 ISEED
c
c   Externals:
c     INTERP    RAN
      REAL*4 RAN
C--

C++
C**********   L M O N O C R M   **********
C
         LOGICAL FUNCTION LMONOCRM(PART, PARAMS, SURF,
     1                             COS_TH, AP, BP, CP, ISEED)
C
C  Determine probability of reflection of the neutron described in PART from
C    a monochromator crystal described by mosaic spread and ideal reflection
C    probabilities given in PARAMS, and whose front surface is defined in SURF.
C    The function will be true if reflection occurs, in which case the
C    effective angle of incidence and surface normal are also returned.
C
C  P.A. Seeger and Mike Fitzsimmons, June 8, 1995.
C
C  Definitions of STRUCTUREs and physical constants:
```

```
      INCLUDE 'mc_geom.inc'
      INCLUDE 'mc_elmnt.inc'
      INCLUDE 'constant.inc'
C
C  Variables in calling sequence:
C    PART = record containing particle coordinates (input)
C    PARAMS = array with parameters describing mosaic crystal (input)
C    SURF = record containing definition of surface (input)
C    COSTH = cosine of angle between PART and effective surface normal (output)
C    AP,BP,CP = direction cosines of the effective surface normal (output)
C    ISEED = random-number generator seed (input/output)
      RECORD /PARTICLE/ PART
      RECORD /SURFACE/  SURF
      REAL*4 PARAMS(*), COS_TH, AP, BP, CP
      INTEGER*4 ISEED
C
C  Externals:
C    ANGLI     NORM      PLNORM    RAN
      REAL*4 PLNORM, RAN
C--

C++
C********** L I G H T R F L **********
C
      LOGICAL FUNCTION LIGHTRFL(COSINC, REFINDX, PROB, ISEED)
C
C  Determine probability of reflection of light at a boundary between
C    media with different indices of refraction.  Arguments are
C    COSINC = the cosine of the angle of incidence, and REFINDX = (index
C    of refraction of new region)/(index of refraction of old region).
C    The two polarizations are averaged, and no estimate of the polarization
C    after reflection is made.  The function is set to .TRUE. if reflection
C    will occur (based on a random number), and .FALSE. if not.  The computed
C    reflection probability PROB is also returned.
C
C  P. A. Seeger, 19 Mar. 1985.  Modified from LREFLCT.
C
C  Variables in calling sequence:
C    COSINC = cosine of angle of incidence (input)
C    REFINDX = relative index of refraction of next region (input)
C    PROB = reflection probability (output)
C    ISEED = random-number generator seed
      IMPLICIT NONE
      INTEGER*4 ISEED
      REAL*4 COSINC, REFINDX, PROB
C
C  External:
C    RAN
      REAL*4 RAN
C--

C++
C********** L R E F L C T **********
C
      LOGICAL FUNCTION LREFLCT(SINPHI, CXRATIO, PROB, ISEED)
C
C  Determine probability of reflection of a neutron at a boundary between
C    media with different coherent scattering lengths.  Arguments are
```

```
C    SINPHI = the sine of the angle from the surface (glancing angle),
C    and CXRATIO = Lambda**2/(2*Pi) times the coherent scattering length
C    ratio (i.e., coherent scattering length per unit volume on the far side
C    of the surface divided by the near side), which is a complex quantity
C    to account for absorption. The function is set to .TRUE. if reflection
C    will occur (based on a random number), and .FALSE. if not.  The computed
C    reflection probability PROB is also returned.
C
C  P. A. Seeger, 1984.
C    07 Jan 1985: Test for valid SINPHI [PAS]
C
C  Variables in calling sequence:
C    SINPHI = sine of glancing angle (cosine of angle of incidence) (input)
C    CXRATIO = 1 - relative index of refraction of next region (input)
C    PROB = reflection probability (output)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      INTEGER*4 ISEED
      REAL*4    SINPHI, PROB
      COMPLEX   CXRATIO
C
C  External:
C    RAN
      REAL*4 RAN
C--

C++
C********** M O S A I C **********
C
      SUBROUTINE MOSAIC(PART, SURF, ROTX, ROTY, ROTZ,
     1                  AP, BP, CP, COSTH, ISEED)
C
C  Compute the cosine of the angle of incidence of the particle PART with
C    respect to surface SURF.  The surface normal vector is rotated about each
C    axis by amounts selected from normal distributions of widths ROTX, ROTY,
C    and ROTZ; the angles are assumed small enough that the sine is equal to
C    the angle and the cosine is one.  The resulting (renormalized) effective
C    surface vector is returned as (AP, BP, CP).  The cosine is returned in
C    COSTH.  No test is made to assure that particle is actually on the surface.
C
C  P. A. Seeger, 27 January, 1995.
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input)
C    SURF = record with surface definition parameters (input)
C    ROTX,ROTY,ROTZ = rms of sines of mosaic spread angles (input)
C    AP,BP,CP = direction cosines of effective surface normal (output)
C    COSTH = cosine of angle between particle and surface normal (output)
C    ISEED = random-number generator seed (input/output)
      RECORD /PARTICLE/ PART
      RECORD /SURFACE/ SURF
      REAL*4 ROTX, ROTY, ROTZ, AP, BP, CP, COSTH
      INTEGER*4 ISEED
C
```

```
C  Externals
C    ANGLI     NORM      PLNORM
      REAL*4 PLNORM
C--

C++
C********** M O V E X **********
C
      SUBROUTINE MOVEX(PART, EXDIST)
C
C  Advance the position of particle PART by a distance EXDIST, including
C    gravitational droop.  If the distance is insignificantly small, a very
C    small step is made in the coordinate with the largest velocity.  The
C    position and time of PART are updated, and EXDIST is set to 0 on return.
C
C  P. A. Seeger, 1980.
C    Modified for time dependence, 1984.
C    08 Feb 1994: converted from COMMON to STRUCTUREs; added PART to calling
C                 sequence; include gravitational droop  [PAS]
C    19 Apr 1994: make sure particle has actually moved  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
      INCLUDE 'constant.inc'
C
C  Variables in calling sequence:
C    PART = record containing particle coordinates (input/output)
C    EXDIST = distance to move particle (input/output)
      RECORD /PARTICLE/ PART
      REAL*4 EXDIST
C
C  No externals
C--

C++
C********** N E X T R G **********
C
      INTEGER*4 FUNCTION NEXTRG(PART, GEOM, IREG, ISURF)
C
C  Find what region the particle defined in PART, presently on surface
C    ISURF exiting region IREG, will enter on the other side of the surface.
C    Returns current region number (IREG) if not on a surface, and returns 0
C    if no valid region is found from the definitions in GEOM.  No
C    parameters are changed.
C
C  P. A. Seeger, 1980.
C    Restructured by P. A. Seeger, 1984.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; added PART and GEOM
C                 to calling sequence  [PAS]
C    29 Jun 1995: if ISTYPE>10, stay in same region instead of being lost [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input)
```

```
C    GEOM = structure with all surface and region definitions (input)
C    IREG = region particle was in before striking surface (input)
C    ISURF = surface upon which particle in /MC_PART/ is located (input)
      RECORD /PARTICLE/ PART
      RECORD /MC_GEOM/ GEOM
      INTEGER*4 IREG, ISURF
C
C  Externals:
C    TESTIN
      LOGICAL TESTIN
C--

C++
C**********   N O R M   **********
C
      SUBROUTINE NORM(X1, X2, X3)
C
C  Normalize three elements (X1,X2,X3) to unit vector magnitude (if not 0).
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
C
C  Variables in calling sequence:
C    X1,X2,X3 = three components of vector to be normalized (input/output)
      IMPLICIT NONE
      REAL*4 X1, X2, X3
C
C  No externals
C--

C++
C**********   N _ S O U R C E   **********
C**********  G E T _ S P A C E  **********
C
      LOGICAL FUNCTION N_SOURCE(NEUTRON, PARAMS, SURF, GFACCEL_2, NCHOP,
     1                          BADFRAME, ISEED)
C
C                     GET_SPACE(PARAMS, PHASPACE, LETHARGY)
C
C  Find source NEUTRON for Monte Carlo, randomly selected in phase space
C    defined in PARAMS, on the specified surface, with velocity in the positive
C    Z-direction.  (If the surface in the first call is a plane perpendicular
C    to the Z-axis, is is NOT checked on subsequent calls.) Function is .TRUE.
C    if neutron source point is within limits of moderator surface and energy,
C    .FALSE. otherwise.  Account for possible gravity-focus device and/or
C    chopper frequency less than beam  repetition rate.  A separate entry
C    (GET_SPACE) is provided to return the phase space volume for normalization.
C
C  P. A. Seeger, May 24, 1994
C    02 Jul 1994: return neutron weight, even if function is .false. [PAS]
C    05 Jan 1995: MC_ELMNT.INC now has parameter offsets, not UNIONs [PAS]
C    10 Jan 1995: use PLNRMTBL for log(EN/1meV) instead of PLENEFF; pass
C                 parameters to revised PLTIME; convolute pulse width [PAS]
C    16 Feb 1995: new calling sequence for PLNRMTBL [PAS]
C    07 Mar 1995: allow rectangular and off-center phase space apertures [PAS]
C    22 Mar 1995: omit TOF if parameter negative; omit table lookup if length 1
C                 or less; allow triangular (velocity selector) wavelength [PAS]
C    28 Mar 1995: source surface in calling sequence, Z not necessarily 0 [PAS]
```

```
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
      INCLUDE 'mc_elmnt.inc'
      INCLUDE 'constant.inc'
C
C  Variables in calling sequence:
C    NEUTRON = record with coordinates of neutron (output)
C    PARAMS = array with source parameters (input)
C    SURF = record with surface definition (input)
C    GFACCEL_2 = half the acceleration of 2nd aperture (m/us/us, input)
C    NCHOP = chopping ratio, period of chopper / period of beam (input)
C    BADFRAME = flag that neutron belongs to a different proton pulse (output)
C    ISEED = random-number generator seed (input/output)
C    PHASPACE = volume of phase space sampled (m**2-ster)
C    LETHARGY = number of lethargy units of energy space sampled
      RECORD    /PARTICLE/ NEUTRON
      RECORD    /SURFACE/  SURF
      REAL*4    PARAMS(*), GFACCEL_2, PHASPACE, LETHARGY
      INTEGER*4 NCHOP, ISEED
      LOGICAL   BADFRAME, GET_SPACE
C
C  Externals:
C     DIST      MOVEX     PLNRMTBL  PLTIME     RAN       RNDCRCL
      REAL*4 DIST, PLNRMTBL, PLTIME, RAN
C--

C++
C********** O P E R A T E **********
C********** E X I T _ R E G **********
C
      SUBROUTINE OPERATE(PART, EXDIST, PARAMS, GEOM, IREG, JSURF,
     1                   NAME, FLAG, PART_2, DET_WT, IX, IY, ISEED)
C
C               EXIT_REG(PART, GEOM, IREG, JSURF)
C
C  Routines to operate on a particle within (or exiting from) a region
C    containing material, collimation elements, time-dependent devices,
C    samples, or detectors.  Included region types/actions are
C         1    amorphous unpolarized material: move to exit w/reduced weight
C         2    aluminum: move to exit w/reduced weight
C         5    beryllium: move to exit w/reduced weight
C         6    single-crystal filter: move to exit w/reduced weight
C        11    multi-slit collimator, vertical: selects subregion w/o moving
C        12    multi-slit collimator, horizontal: selects subregion w/o moving
C        13    crystal monochromator: reflect from surface or move to exit, and
C              rotate coordinates
C        20    blade or disk chopper: move to exit OR set weight=0
C        22    gravity focuser: selects subregion without moving
C        23    removable beamstop: set weight=0, secondary moves to exit
C        30    fixed-Q scatterer: transmitted or scattered particle moves to
C              exit with modified weight
C        31    hard-sphere scatterer: transmitted or scattered particle moves
C              to exit with modified weight
C        32    isotropic scatterer with constant energy change: transmitted or
C              scattered particle is moved to exit with modified weight
C        34    inelastic scatter using S(alpha,beta) from MCNP: transmitted or
C              scattered particle is moved to exit with modified weight
```

```
C        35    reflectometry, multilayer: reflect from surface w/o moving,
C              same weight if transmitted or reduced weight if scattered
C        36    general powder scatterer: transmitted or scattered particle moves
C              to exit with modified weight
C        40    single detector: determine detection probability
C        41    linear detector, vertical: determine y-bin and probability
C        42    linear detector, transverse: determine x-bin and probability
C        43    2-dimensional detector: determine x- and y-bins and probability
C        44    linear detector, longitudinal: determine x-bin and probability
C        90    source size and phase space: weight=0 if outside surface
C
C  P. A. Seeger, April 20, 1994.
C    17 May 1994: corrected detector encoding for less-than-minimum [PAS]
C    24 May 1994: added FLAG to calling sequence [PAS]
C    05 Jan 1995: MC_ELMNT.INC now has parameter offsets, not UNIONs [PAS]
C    01 Feb 1995: added types 5 (Be), 13 (crystal monochromator), and 35
C                 (reflectometry) [PAS]
C    17 Feb 1995: added type 31 (hard spheres); more SAVEd variables [PAS]
C    02 Mar 1995: use negative JSURF as flag for reflection [PAS]
C    06 Mar 1995: type 31: compute abs(KZ), return trans & refl neutrons [PAS]
C    09 Mar 1995: types 33 (Ni powder), 44 (longitudinal detector) [PAS]
C    17 Mar 1995: change POWDER to Ni_POWDR; change argument of PLEXP in
C                 estimating multiple scattering [PAS]
C    01 Apr 1995: added NAME to calling sequence, type 34 (inelastic) [PAS]
C    13 Apr 1995: no time binning, eliminate IT from calling sequence; fix
C                 factors of 2 in calls to ELSCAT for types 33, 34 [PAS]
C    08 May 1995: correct errors in multiple scattering [JAG,PAS]
C    25 May 1995: still was error in inelastic multiple scattering [PAS]
C    07 Jun 1995: revised handling of monochromator type 13 [MFitz,PAS]
C    20 Jun 1995: added subtypes 20.n, disk chopper moving vertically and/or
C                 counter-rotating [PAS]
C    14 Jul 1995: add type 36, powder scatterer [PAS,LLD,UCW]
C    04 Aug 1995: add type 6, single-crystal filter [PAS]
C    26 Aug 1995: add type 32, isotropic scatterer; omit type 33 [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
      INCLUDE 'mc_elmnt.inc'
      INCLUDE 'constant.inc'
C
C  Variables in calling sequence:
C    PART = record containing description of particle (input/output)
C    EXDIST = distance to exit surface particle is aimed at (m) (input/output)
C    PARAMS = array with description of what's in the region (input)
C    GEOM = structure with all surface and region definitions (input)
C    IREG = region number of device, or subregion within device (input/output)
C    JSURF = surface number, if particle is on surface (input/output)
C    NAME = name of region, used as file name for type 34 (input)
C    FLAG = flag set to .FALSE. if (e.g.) chopper in wrong frame (output)
C    PART_2 = description of particle created by operation (output)
C    DET_WT = statistical weight of detected particle (output)
C    IX, IY = position bin numbers of detected particle (output)
C    ISEED = random-number generator seed (input/output)
      RECORD    /PARTICLE/ PART, PART_2
      RECORD    /MC_GEOM/  GEOM
      REAL*4    PARAMS(*), EXDIST, DET_WT
```

```
      CHARACTER NAME*40
      LOGICAL   FLAG
C
C  Externals:
C    ANGLI     ATTEN_A1  ATTEN_Be  ATTEN_X   DTOEX     ELSCAT    GRAV_FOC
C    KERNEL    LMONOCRM  MOVEX     NEXTRG    ORRAND    PLEXP     PLNORM
C    PLQSPHR   POWDER    RAN       REFLAYER  RFLN      TESTIN    XCHOPPER
      REAL*4  ATTEN_A1, ATTEN_Be, ATTEN_X, GRAV_FOC, PLEXP, PLNORM,
     1        PLQSPHR, RAN, REFLAYER, XCHOPPER
      INTEGER NEXTRG
      LOGICAL LMONOCRM, TESTIN
C--

C++
C********** O P T I C S **********
C
      SUBROUTINE OPTICS(PART, GEOM, IREG, ISURF, JREG,
     1                  RATIO, BETA, ISEED)
C
C  Solve for probability of reflection, refraction, or diffuse reflection
C    for photon PART passing from region IREG across surface ISURF to region
C    JREG.  Surface parameters are found in GEOM, and the position and
C    direction in PART are updated.  The ratio of new index of refraction
C    to old is RATIO.  At exit, JREG will be the new region number for the
C    particle: IREG if reflected, JREG if transmitted, and 0 if lost or
C    absorbed.
C
C  P. A. Seeger, 13 Mar. 1985.
C    02 Apr 1985:  Changed diffuse reflection to cosine law by calling WOBBLE
C                  [PAS]
C    05 Apr 1985: Changed diffuse surfaces to include transmission as well as
C                 reflection [PAS]
C    11 Jun 1987: Added BETA to calling sequence (was always 0.9) [PAS]
C    03 Feb 1994: converted from COMMON to STRUCTUREs; added PART and GEOM
C                 to calling sequence  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input/output)
C    GEOM = structure with all surface and region definitions (input)
C    IREG = current region (input)
C    ISURF = surface particle is on (input)
C    JREG = region across ISURF at the particle location; region particle
C           will be in after interacting at surface (input/output)
C    RATIO = (new index of refraction)/(old index of refraction) (input)
C    BETA = surface roughness parameter for use in call to WOBBLE (input)
C    ISEED = random-number generator seed (input/output)
      RECORD /PARTICLE/ PART
      RECORD /MC_GEOM/ GEOM
      REAL*4 RATIO, BETA
      INTEGER*4 IREG, ISURF, JREG, ISEED
C
C  Externals:
C    ANGLI,  LIGHTRFL,  RFLN,    SNELL,   WOBBLE
      LOGICAL LIGHTRFL
```

```
C--

C++
C********** O R R A N D **********
C
      SUBROUTINE ORRAND(VX, VY, VZ, ISEED)
C
C  Generate random orientation unit vector (VX,VY,VZ), in 4 pi
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090;
C    new algorithm, P.A. Seeger, 1980.
C    11 Mar 1995: use RAN0 to desequentialize random numbers [PAS]
C
C  Variables in calling sequence:
C    VX, VY, VZ = direction cosines (output)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      INTEGER*4 ISEED
      REAL*4 VX,VY,VZ
C
C  Externals
C    RAN0
      REAL*4 RAN0
C--

C++
C********** P L C N V L **********
C
      REAL*4 FUNCTION PLCNVL(N, WIDTHS, ISEED)
C
C  Sample a distribution generated by a convolution of N square
C    distributions with individual full widths given in the array
C    WIDTHS.  E.g., if N = 2 the distribution is trapezoidal,
C    becoming triangular if WIDTHS(1) = WIDTHS(2).
C
C  P. A. Seeger, 3 Jan. 1985.
C    11 Mar 1995: use RAN0 to desequentialize random numbers [PAS]
C
C  Variables in calling sequence:
C    N = number of square distributions to be convoluted (input)
C    WIDTHS(N) = array of full widths of square distributions (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      INTEGER*4 N, ISEED
      REAL*4 WIDTHS(N)
C
C  Externals:
C    RAN0
      REAL*4 RAN0
C--

C++
C********** P L E X P **********
C
      REAL*4 FUNCTION PLEXP(ALPHA, PLMAX, ISEED)
C
C  Sample a random distribution of the form exp(-ALPHA*X), on the range
```

```
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
C    Restructured by P.A. Seeger, 1984.
C    23 Jul 1994: restructured to avoid ALOG(0); test for ALOG(0);
C                 infinite range if PLMAX is zero [PAS]
C    16 Feb 1995: revise limits for using EXP to range 0.0002 - 17;
C                 protect against ALPHA=0 [PAS]
C
C  Variables in calling sequence:
C    ALPHA = 1/e factor of exponential distribution (input)
C    PLMAX = upper limit on distribution (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      INTEGER*4 ISEED
      REAL*4 ALPHA, PLMAX
C
C  Externals:
C    RAN
      REAL*4 RAN
C--

C++
C********** P L N O R M **********
C
      REAL*4 FUNCTION PLNORM(XBAR, SIG, ISEED)
C
C  Sample a normal distribution of mean XBAR and standard deviation SIG
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
C    Restructured by P.A. Seeger, 1984.
C    11 Mar 1995: use RAN0 to desequentialize random numbers [PAS]
C
C  Variables in calling sequence:
C    XBAR = mean of normal distribution (input)
C    SIG = standard deviation of normal distribution (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT  NONE
      INTEGER*4 ISEED
      REAL*4    XBAR, SIG
C
C  Externals:
C    RAN0
      REAL*4 RAN0
C--

C++
C********** P L N R M T B L **********
C
      REAL*4 FUNCTION PLNRMTBL(TABLE, NDATA, CENTER, SIGMA, ISEED)
C
C  Select a random value from a TABLE of length NDATA, whose entry points are
C    equally spaced cumulative normal-curve values centered at CENTER in the
C    table with standard deviation SIGMA.
C
C  P. A. Seeger, Jan. 10, 1995.  Copied from PLENEFF with table in calling
C    sequence instead of built in.
C    13 Feb 1995: CENTER and SIGMA in calling sequence; limits on IR [PAS]
C
```

```
C  Variables in calling sequence
C    TABLE = cumulative data points (MUST be monotonic increasing) (input)
C    NDATA = number of entries in data table (input)
C    CENTER = position in table of center of normal curve (input)
C    SIGMA = standard deviation of normal curve in table index units (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT  NONE
      INTEGER   NDATA
      REAL*4    CENTER, SIGMA, TABLE(*)
      INTEGER*4 ISEED
C
C  Externals
C    PLNORM
      REAL*4 PLNORM
C--

C++
C**********   P L P O I S S N   **********
C
      REAL*4 FUNCTION PLPOISSN(XM, ISEED)
C
C  Returns a floating-point integer value that is a random deviate drawn from
C    a Poisson distribution of mean XM.  Use the desequentialized random-
C    number generator, RAN0.
C
C  Press et al., "Numerical Recipes" (2nd ed., Cambridge, 1992) p. 283
C    18 Mar 1995: adapted; protect against negative or zero XM [PASeeger]
C
C  Variable in calling sequence:
C    XM = mean of Poisson distribution to be sampled (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT  NONE
      REAL*4    XM
      INTEGER*4 ISEED
C
C  Externals:
C    GAMMLN    RAN0
      REAL*4 GAMMLN, RAN0
C--

C++
C********** P L Q S P H R **********
C
      REAL*4 FUNCTION PLQSPHR(RADIUS, ISEED)
C
C  Select a random momentum transfer Q from the scattering law for a
C    hard sphere of the specified RADIUS.  Units of Q are the inverse of
C    the units of RADIUS.  Probabilities are found from tables of the
C    cumulative distribution of the Rayleigh scattering law.
C
C  P. A. Seeger, 1984.
C  31 May 1985: corrected distribution for solid angle: cumulative distribution
C               is F = 1 - sin(X)*sin(X)/X**4 + 2*sin(X)*cos(X)/X**3 - 1/X**2
C               [A.T.Boothroyd, RAL]
C  21 Jul 1985: Series expansion for F < 0.05 [PAS]
C  06 Mar 1995: even simpler expansion for F < 1.e-5 [PAS]
C
```

```
C  Calculate the probability of reflection of a neutron from any number of
C    layers on top of a substrate.  The wavevector component perpendicular
C    to the surface is KZERO, and the layers are defined in PARAMS.  In case
C    of error, the probability is set to -1.  If the external medium is not
C    void, its value of 4pi*n*b should be subtracted from the value for each
C    layer.
C
C  Greg Smith 1/30/95
C    07 Feb 1995: made a FUNCTION; put parameters in calling sequence [PAS]
C
C  Include file with parameter list definitions
      IMPLICIT  NONE
      INCLUDE 'mc_elmnt.inc'
C
C  Variables in calling sequence:
C    KZERO = perpendicular component of wavevector (1/A)
C    PARAMS = vector containing definitions of layers
      REAL*4    KZERO, PARAMS(*)
C
C  No externals
C--

C++
C**********  R F L N  **********
C
      SUBROUTINE RFLN(PART, COSTH, AP, BP, CP)
C
C  Perform reflection of particle PART, with angle of reflection
C    equal to angle of incidence (cosine of angle = COSTH, measured with
C    respect to surface normal AP,BP,CP).  Sign of COSTH is changed to
C    represent new direction of particle.
C
C  M. W. Johnson, Rutherford and Appleton Laboratories report RL-80-065.
C    Modified calling sequence, P. A. Seeger, 1984.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; add PART to calling
C                 sequence [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates
C    COSTH = cosine of angle of incidence (input)
C    AP,BP,CP = unit vector in direction of normal to surface (input)
      RECORD /PARTICLE/ PART
      REAL*4 COSTH, AP, BP, CP
C
C  No externals
C--

C++
C********** R N D C R C L **********
C
      SUBROUTINE RNDCRCL(X, Y, ISEED)
C
C  Generate random point (X,Y) within a unit circle
C
```

```
C    RADIUS = radius of the hard sphere (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      INTEGER*4 ISEED
      REAL*4 RADIUS
C
C  Externals
C    RAN
      REAL*4 RAN
C--

C++
C********** P L T I M E **********
C
      REAL*4 FUNCTION PLTIME(E, PARAMS, ISEED)
C
C  Select a random time of neutron emission, from a distribution appropriate
C    to energy E (eV), described by values in PARAMS.  The distribution is a
C    sum of an epithermal exponential and a thermal term (linear combination of
C    two exponentials) convoluted with a Gaussian.  The epithermal exponential
C    and the Gaussian constants are proportional to lambda.  The switch
C    function between the epithermal and thermal terms is an exponential
C    of a power of a wavelength ratio.
C
C  P. A. Seeger, 1984.
C    10 Jan 1995: all parameters of the distribution provided externally;
C                 new switch function [PAS]
C    13 Feb 1995: no upper limit to exponential; revised PARAMS [PAS]
C    27 Mar 1995: protect against missing arguments [PAS]
C    05 Sep 1995: third exponential term; revised PARAMS [PAS]
C
C  Include files with physical constants and parameter list definitions
      IMPLICIT NONE
      INCLUDE 'constant.inc'
      INCLUDE 'mc_elmnt.inc'
C
C  Variables in calling sequence
C    E = neutron energy (eV) (input)
C    PARAMS = block containing parameters defining the lineshape
C    ISEED = random-number generator seed
      INTEGER*4 ISEED
      REAL*4 E, PARAMS(*)
C
C  Externals
C    RAN,   PLEXP,   PLNORM
      REAL*4 RAN, PLEXP, PLNORM
C--

C++
C********** P L T R N G L **********
C
      REAL*4 FUNCTION PLTRNGL(XBAR, FWHM, ISEED)
C
C  Sample a triangular distribution of mean XBAR and full width
C    at half maximum of FWHM.
C
C  P. A. Seeger, 1984.  Revised 4 Jan. 1985.
```

```
C  Variables in calling sequence:
C    XBAR = center of triangular distribution (input)
C    FWHM = full width at half maximum (input)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      REAL*4 XBAR, FWHM
      INTEGER*4 ISEED
C
C  Externals:
C    RAN
      REAL*4 RAN
C--

C++
C**********   P O W D E R   **********
C
      SUBROUTINE POWDER (PARAMS, LAMBDA, SINTH, SIGSCAT, SIGABS, ISEED)
C
C   Scattering kernel for isotropic polycrystalline (powder) sample.
C     Given a powder defined by parameters in PARAMS and the neutron wavelength
C     LAMBDA, returns the sine of the Bragg angle of the neutron from a randomly
C     selected reflection, and the macroscopic scattering and absorption cross
C     sections SIGSCAT and SIGABS.
C
C  From Uli Wildgruber and Larry Passell, 14 July, 1995; PASeeger.
C
C  Variables in calling sequence:
C    PARAMS  = array with scattering cross section parameters (input)
C    LAMBDA  = neutron wavelength (A) (input)
C    SINTH   = sin(THETA) for chosen reflection (output)
C    SIGSCAT = macroscopic scattering cross section (/cm) (output)
C    SIGABS  = macroscopic absorption cross section (/cm) (output)
C    ISEED   = Random number seed (input/output)
      IMPLICIT  NONE
      INCLUDE   'mc_elmnt.inc'
      INCLUDE   'constant.inc'
      REAL*4    PARAMS(*), LAMBDA, SINTH, SIGSCAT, SIGABS
      INTEGER*4 ISEED
C
C  Externals:
C    HUNT      RAN
      REAL*4 RAN
C--

C++
C**********     R A N     **********
C
      REAL*4 FUNCTION RAN(ISEED)
C
C  Random number generator, linear congruential method
C
C  Press et al., "Numerical Recipes" (Cambridge, 1986) section 7.1
C
C  Variables in calling sequence:
C    ISEED = 32-bit integer seed value (input/output)
      IMPLICIT  NONE
      INTEGER*4 ISEED
C
```

```
C  No externals
C--

C++
C**********    R A N 0    **********
C
      REAL*4 FUNCTION RAN0(ISEED)
C
C  Random number generator with shuffle table to reduce sequential correlations
C
C  Press et al., "Numerical Recipes" (Cambridge, 1986) section 7.1
C
C  Variables in calling sequence:
C    ISEED = 32-bit integer seed value (input/output)
       IMPLICIT  NONE
       INTEGER*4 ISEED
C
C  External:
C    RAN
       REAL*4  RAN
C--

C++
C********** R E A D _ 1 D **********
C
      SUBROUTINE READ_1D(IUNIT,TYPE,TITLE,FILEID,PFLAG,PARAMS,NP,
     ,                   HEADER,NH,X,DX,Y,DY,IFLAG,IERROR)
C
C     Search input device IUNIT for the next data file of type TYPE (or next
C  file of any one-dimensional type if TYPE is blank when called).  Return all
C  header information and data arrays X (which is points if the input value of
C  PFLAG is 'P' or 'p', or bin boundaries otherwise), DX (which is either bin
C  normalization if the output value of IFLAG is 0, or the rms of the points
C  in the bin if IFLAG = 1), Y, and DY (standard deviation).  IERROR will be 0
C  if a suitable  data file was found and successfully read, -1 if none found,
C  and positive if a read error occurred.
C
C     The structure of a 1-D file is:
C  One formated record (A4,1X,A40,1X,A17) containing TYPE, TITLE, and FILEID.
C  A block of fewer than 100 parameters, separated by blanks or commas,
C     and terminated with a '/'.  The first parameter must be the number
C     of channels NCHANS, and if TYPE = 'LOGT' the next four must be the
C     parameters of the logarithmic time scale.  (Additional assignments
C     of parameters beyond the fifth are assumed by other programs.)  The
C     parameters are not reinitialized; they default to values at input.
C  A string of up to 720 characters identifying all raw data used and any
C     operations performed.  There are up to nine 80-character records.
C     The first blank character (after character 1) terminates the string.
C  One or more "blocks", including one "DATA" block:
C     The first record of a block is 'POINTS', 'BINS', 'AREA', 'RMSBIN',
C        'DATA', or 'STDDEV'.
C     Blocks contain NCHANS (or NCHANS+1 for a BINS block) numbers,
C        separated by blanks or commas, terminated with a '/'.
C     If there is no POINTS or BINS block, the bins will default to the
C        log time scale if TYPE = 'LOGT'; computed linear, logarithmic, or
C        log/lin/log if appropriate parameters exist in the parameter block;
C        or integers otherwise.
C     The STDDEV block (if any) containing the statistical standard deviations
```

```
C       of the data must come after the data block.  If omitted, the defaults
C       will be the square roots of the data.
C    There may be either an 'AREA' block containing a normalizing function
C       to be divided into the data, or an 'RMSBIN' block giving the rms
C       of the X-values used to average within the bin.
C
C    Structures other than 1-D files (e.g., 2-D files) may be interspersed
C  with 1-D files, but must end with a blank record.
C
C  P. A. Seeger, Los Alamos National Laboratory, Jan. 16, 1988.
C    Dec. 8, 1988: changed name (was READ1D); added DX and IFLAG to calling
C                  sequence; read "AREA" or "RMSBIN" blocks.
C    Aug. 9, 1989: replace UAHR in calling sequence with PFLAG to return points
C                  instead of bins; compute Q-bins from parameters.
C    Oct. 3, 1989: default deviations to 0 if STDDEV block present.
C    Jan. 16, 1990: change types 'LOGT' or '<T  ' to 't   '
C    Oct. 18, 1990: revert to 'LOGT' if using parameters to compute bins
C    June 19, 1991: change PFLAG to in/out; if blank at input, set to 'P' or 'B'
C                  depending on what is read from file
C    Aug. 14, 1992: set PARAMS(18)=3 instead of TYPE='LOGT' for computed bins
C
C  Variable in calling sequence:
C    IUNIT   I       Input     Fortran unit number to be searched and read
C    TYPE    C*4     In/Out    Type of data file to search for, or type found
C    TITLE   C*40    Output    Title from first record of file
C    FILEID  C*17    Output    Instrument_Date_Time identifier
C    PFLAG   C*1     In/Out    Conversion flag: 'P' or 'p' to return X as
C                              points, or ' ' to return whatever was recorded
C                              in file and then set to 'B' or 'P'
C    PARAMS  R(*)    Output    Parameters saved with the data file;
C                              PARAMS(1) = NCHANS = number of channels
C                              PARAMS(2..5) = log time parameters
C                              PARAMS(18..23) = Q-binning parameters
C    NP      I       Input     Number of PARAMS to be returned to caller
C    HEADER  C(9)*80 Output    Character string describing operations
C                              previously performed on data
C    NH      I       Output    Number of characters in HEADER
C    X       R(*)    Output    Bin boundaries (NCHANS+1 numbers), or points
C                              (NCHANS numbers) if PFLAG is 'P' or 'p'.
C    DX      R(*)    Output    Normalizing factor (area) of each bin, or
C                              rms of points used in average (see IFLAG)
C    Y       R(*)    Output    Data values summed or averaged over bins
C    DY      R(*)    Output    Standard deviations of Y
C    IFLAG   I       Output    0 if DX is normalizing factor, 1 if rms, or
C                              -1 if neither was read from file
C    IERROR  I       Output    0 for successful read, -1 for end-of-file,
C                              or system-dependent message number
C
C  No Externals
C-

C++
C********** R E A D _ 2 D **********
C
      SUBROUTINE READ_2D(IN, TYPE, TITLE, FILEID, PFLAG, PARAMS, NP,
     1                   HEADER, NH, X, DX, Y, DY, NY,
C**MS Fortran requires [HUGE] attribute on Z and DZ
     2                   Z, DZ, NXZ, IFLAG, IERROR)
```

```
C
C     Search input device IN for the next data file of type TYPE (or next file
C  of any two-dimensional type if TYPE is blank when called). Return all header
C  information, the one dimensional arrays X, DX, Y, and DY, and the 2D data
C  arrays Z(X,Y) and DZ(X,Y), stored in arrays with first dimension NXZ. The
C  X and Y arrays are normally bin boundaries, but may be returned as points by
C  setting PFLAG; if "PX" then X will be points, if "PY" then Y will be points,
C  and if "P" both. The numbers of X and Y bins are always read from the header
C  (1st and 24th entries in PARAMS). The input NY may be reduced to the value
C  in the data file. DX and DY may be either normalizing functions (areas of
C  bins) or rms deviations of points included in bin averages (flagged by the
C  value returned in IFLAG: 1 if DX is rms, 2 if DY is rms, 3 if both, -1 if
C  no such blocks were read). DZ contains the standard deviations of Z, either
C  as read from the input file or assuming Poisson statistics. At exit,
C  IERROR will be 0 if a suitable data file was found and successfully read,
C  -1 if no file found, and positive if a read error occurred.
C
C     The structure of a 2D file is:
C  One formated record (A4,1X,A40,1X,A17) containing TYPE, TITLE, and FILEID
C  A block of fewer than 100 parameters, separated by blanks or commas,
C     and terminated with a '/'. The first parameter must be the number
C     of channels NX in the X direction. (Additional assignments of parameters
C     are listed below.) The parameters are not reinitialized; they default
C     to values at input.
C  A string of up to 720 characters identifying all raw data used and any
C     operations performed. There are up to nine 80-character records.
C     The first blank character (after character 1) terminates the string.
C  One or two optional blocks, either "POINTS" or "BINS", the first defining
C     the X-axis and the second (if any) defining the Y-axis. If the blocks
C     are not present, X (or Y) will default to the log t scale if the first
C     (or any subsequent) letter of TYPE is 'T', or to integer steps
C     otherwise.
C  One or two optional "AREA" or "RMSBIN" blocks, containing normalization
C     factors (if the keyword is "AREA") to be divided into the raw histogram
C     counts, or else the rms average ("RMSBINS") of the values of X or Y used
C     in computing Z averages. The first such block refers to X, and the
C     second (if any) to Y. By using "AREA" blocks, Z may be stored in the
C     unnormalized form to preserve its Poisson statistics. These blocks are
C     returned to the caller as DX and DY, which will be set to unity if the
C     blocks are not present.
C  Any number (at least one!) of rows of Z:
C     The first record of a row block has a '=' within the first 4 characters
C        and a value of IY anywhere in the 15 characters following. The value
C        of IY must be between 1 and NY, or within the time-slice limits given
C        in the header if Y is time.
C     A block of NX values, separated by blanks or commas, terminated
C        with a '/'.
C  Each row block may be immediately followed by a "STDDEV" block containing
C     NX values of the standard deviation of Z for that row; otherwise Z will
C     be assumed Poisson and the standard deviation will be the square root.
C
C     Structures other than 2D files (e.g., 1D files) may be interspersed
C  with 2D files, but must end with a blank record.
C
C     Definintions of the parameter block are
C        PARAM(1)  = NX, number of points in a block, or bins in X-direction
C        PARAM(2)  = DTOVERT, dt/t for log time scale
```

```
C         PARAM(4)  = TSTART, beginning of first time slice
C         PARAM(5)  = TZERO, time delay of detector
C         PARAM(6)  = maximum time, last bin boundary
C         PARAM(7)  = size of detector element in x-direction (mm)
C         PARAM(8)  = size of detector element in y-direction (mm)
C         PARAM(9)  = x of beam centerline (detector units)
C         PARAM(10) = y of beam centerline (detector units)
C         PARAM(11) = power of the units of Z
C         PARAM(12) = source-to-sample distance (m)
C         PARAM(13) = sample-to-detector distance (m)
C         PARAM(14) = collimation type (1, 5, or 7)
C         PARAM(15) = minimum detector radius included; x semiaxis if elliptical
C         PARAM(16) = IFIRST, first time slice included
C         PARAM(17) = last time slice included
C         PARAM(18) = Q-bin type: 0 = linear, 1 = logarithmic, 2 = log/lin/log
C         PARAM(19) = switch point from log to linear
C         PARAM(20) = switch point from linear back to log
C         PARAM(21) = number of bins between Q1 and Q2
C         PARAM(22) = minimum Q bin boundary
C         PARAM(23) = maximum Q bin boundary
C         PARAM(24) = NY, number of blocks of data, or bins in Y-direction
C         PARAM(25) = integrated beam current in uA-hr
C         PARAM(26) = resolution width factor, dR/R < (this factor)*dQ/Q
C         PARAM(27) = detector efficiency at 1 A
C         PARAM(28) = scattering angle to a fixed detector (degrees)
C         PARAM(29) = incident energy, direct-geometry inelastic (meV)
C         PARAM(30) = final energy, inverted-geometry inelastic (meV)
C         PARAM(31) = y semiaxis of elliptical mask (cf. PARAM(15))
C
C  P. A. Seeger, Los Alamos National Laboratory, Jan. 17, 1988.
C    02 Nov 1988: add DZ to calling sequence; read STDDEV blocks.
C    08 Nov 1988: determine NY; set bins to 0 before reading.
C    08 Dec 1988: renamed (was READ2D); revised calling sequence!!!
C                 allows RMSBIN blocks.
C    21 Dec 1988: IFLAG=-1 if no AREA or RMSBINS blocks
C    12 Aug 1989: use new header index information; replace UAHR in calling
C                 sequence with PFLAG for points instead of bins
C    25 Sep 1989: offset IY if Y is time and IFIRST is given in header
C    05 Jul 1992: also offset IY if Y is wavelength
C    24 Jun 1994: MS Fortran version recompiled with "HUGE" attributes
C    13 Mar 1995: integer BINS if input block was null [PAS]
C    29 Mar 1995: restructure to speed up when dimensions large; require
C                 PARAM(24); allow 't' as well as 'T' in TYPE [PAS]
C    06 Apr 1995: eliminate BACKSPACE when reading '=' blocks [JAG,PAS]
C
C  Variable in calling sequence:
C    IN      I       Input     Fortran unit number to be searched and read
C    TYPE    C*4     In/Out    Type of data file to search for, or type found
C    TITLE   C*40    Output    Title from first record of file
C    FILEID  C*17    Output    Instrument_Date_Time identifier
C    PFLAG   C*2     Input     Flag to return points instead of bins;
C                              'PX' for X, 'PY' for Y, or 'P' for both
C    PARAMS  R(*)    In/Out    Parameters saved with the data file;
C                              PARAMS(1) = NX = number of channels
C                              PARAMS(2..5) = log time parameters
C    NP      I       Input     Number of PARAMS to be returned to caller
C    HEADER  C(9)*80 Output    Character string describing operations
C                              previously performed on data
```

```
C    NH       I        Output    Number of characters in HEADER
C    X        R(*)     Output    Bin boundaries (NX+1 numbers)
C    DX       R(*)     Output    Normalization factor to be divided into Z
C                                 as a function of its first index, OR rms of
C                                 bin (see IFLAG)
C    Y        R(*)     Output    Bin boundaries for the second index of Z
C    DY       R(*)     Output    Normalization factor to be divided into Z
C                                 as a function of its second index, OR rms of
C                                 bin (see IFLAG)
C    NY       I        In/Out    Maximum number of Y blocks allowed/returned
C    Z        R(NXZ,*) Output    2D data array
C    DZ       R(NXZ,*) Output    Standard deviations of Z
C    NXZ      I        Input     First dimension of Z and DZ in calling program
C    IFLAG    I        Output    1 if DX is rms widths of X-bins, 2 if DY is
C                                 rms width of Y-bins, 3 if both, 0 if neither;
C                                 -1 if no blocks were read
C    IERROR   I        Output    0 for successful read, -1 for end-of-file,
C                                 or system-dependent message number
C
C  No Externals
C-

C++
C********* R E A L O U T **********
C
      SUBROUTINE REALOUT(Q,N,IUNIT)
C
C  Output an array of N real*4 numbers Q to unit IUNIT, as ASCII characters
C    in free format with between 5-1/2 and 7 significant digits, separated
C    by commas, with 126 or fewer characters per record.  Each unit record
C    begins with a blank, and the entire array is terminated with a '/'.
C
C  P. A. Seeger, Los Alamos National Laboratory, April 9, 1987.
C    30 Sep 1987: Use "," instead of "0," for zero.
C    24 Nov 1987: Changed lower limit for F format.
C    05 Dec 1987: Decreased digits in F format by one.
C    02 Dec 1988: Eliminate trailing ",,"; put null after "/".
C    23 Sep 1989: omit final null if buffered line too long.
C    12 Feb 1991: omit "E" and lead "0" in exponent of E format; raise
C                 lower limit for "F" format
C    30 Apr 1992: omit trailing 0s in E format
C    13 Mar 1995: omit null after "/" [PAS]
C    16 Aug 1995: put "E" back (so IDL can read data) [PAS]
C
C  Externals:
C    DIGITS
C
C    Variables in calling sequence
C       Q       R(*)    Input     Array of real numbers to be output
C       N       I       Input     Number of numbers
C       IUNIT   I       Input     Logical unit number for output
C--

C++
C********** R E F L A Y E R **********
C
      REAL*4 FUNCTION REFLAYER(KZERO, PARAMS)
C
```

```
C  P. A. Seeger, 1984.
C    11 Mar 1995: use RAN0 to desequentialize random numbers [PAS]
C
C  Variables in calling sequence:
C    X, Y = random point (output)
C    ISEED = random-number generator seed (input/output)
      IMPLICIT NONE
      REAL*4 X, Y
      INTEGER*4 ISEED
C
C  Externals
C    RAN0
      REAL*4 RAN0
C--

C++
C********** S N E L L **********
C
      SUBROUTINE SNELL(PART, COSTH, AP, BP, CP, RATIO)
C
C  Perform refraction of particle PART, with sin(angle of refraction) =
C    sin(angle of incidence)/RATIO.  If RATIO < 1 and the angle exceeds
C    the critical angle, the particle is reflected.  The surface normal
C    is (AP,BP,CP) and the cosine of the angle of incidence is COSTH;
C    at exit, COSTH is changed to the new angle.
C
C  P. A. Seeger, 13 March, 1985.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; add PART to calling
C                 sequence  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input/output)
C    COSTH = cosine of angle of incidence (input/output)
C    AP,BP,CP = unit vector in direction of normal to surface (input)
C    RATIO = ratio of indices of refraction, new/old (input)
      RECORD /PARTICLE/ PART
      REAL*4 COSTH, AP, BP, CP, RATIO
C
C  External:
C    RFLN
C--

C++
C********** T E S T I N **********
C
      LOGICAL FUNCTION TESTIN(PART, GEOM, IREG, ISURF)
C
C  Tests whether the particle defined in PART is inside region IREG
C    as defined in GEOM.  If ISURF is not 0, then that surface is not
C    included in testing, in order to avoid roundoff/truncation errors
C    when the particle is known to be on the surface.  Function is .TRUE.
C    if inside region IREG, .FALSE. if not.  No parameters are changed.
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
```

```
C    Modified by P.A. Seeger (1980) for case of particle on a bounding
C                surface of the region.
C    Restructured by P.A. Seeger, 1984.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; added PART and GEOM
C                to calling sequence  [PAS]
C    29 Jun 1995: don't test surface if STYPE >10  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input)
C    GEOM = structure with all surface and region definitions (input)
C    IREG = number of region being tested for interiorness (input)
C    ISURF = number of surface that particle PART is on (input)
      RECORD /PARTICLE/ PART
      RECORD /MC_GEOM/ GEOM
      INTEGER*4 IREG, ISURF
C
C  No externals
C--

C++
C********** W H I C H R **********
C
      SUBROUTINE WHICHR(PART, GEOM, IREG)
C
C  Finds which region the particle PART is in (or is entering) by testing
C    all boundaries of all regions defined in GEOM.  On return, IREG = region
C    number if successful, IREG = 0 if not in any defined region, and
C    IREG = -1 if in more than one region.
C
C  M.W. Johnson and C. Stephanou, Rutherford Laboratory report RL-78-090.
C    Restructured by P.A. Seeger, 1984.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; added PART and GEOM
C                to calling sequence  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input)
C    GEOM = structure with definitions of all surfaces and regions (input)
C    IREG = number of region that /MC_PART/ is in (output)
      RECORD /PARTICLE/ PART
      RECORD /MC_GEOM/ GEOM
      INTEGER*4 IREG
C
C  Externals:
C    TESTIN
      LOGICAL TESTIN
C--

C++
C********** W O B B L E **********
```

```
      SUBROUTINE WOBBLE(PART, SURF, BETA, AP, BP, CP, COSTH, ISEED)
C
C  Compute the cosine of the angle of incidence of the particle PART with
C    respect to surface SURF.  A vector of length BETA with random orientation
C    is added to the surface normal vector, and the resulting (renormalized)
C    effective surface vector is returned as (AP, BP, CP).  The cosine is
C    returned in COSTH.  If BETA<0, a completely random surface is assumed,
C    independent of ISURF.  No test is made to assure that particle is
C    actually on the surface.
C
C  Based on M. W. Johnson, Rutherford and Appleton Laboratories report
C    RL-80-065.
C    P. A. Seeger, 2 Apr. 1985.
C    03 Feb 1994: converted from COMMON to STRUCTUREs; changed calling
C                 sequence  [PAS]
C
C  Definitions of STRUCTUREs:
      IMPLICIT NONE
      INCLUDE 'mc_geom.inc'
C
C  Variables in calling sequence:
C    PART = record with particle coordinates (input)
C    SURF = record with surface definition parameters (input)
C    BETA = surface roughness parameter:  0 for smooth, 1 for cosine
C           distribution, < 0 or >> 1 for complete randomness (input)
C    AP,BP,CP = direction cosines of effective surface normal (output)
C    COSTH = cosine of angle between particle and surface normal (output)
C    ISEED = random-number generator seed (input/output)
      RECORD /PARTICLE/ PART
      RECORD /SURFACE/ SURF
      REAL*4 BETA, AP, BP, CP, COSTH
      INTEGER*4 ISEED
C
C  Externals
C    ANGLI,   NORM,    ORRAND
C--

C++
C********** X C H O P P E R **********
C
      REAL*4 FUNCTION XCHOPPER(T, TOPEN, TCLOSE, SIGT, V, PERIOD,
     1                         LOPENING, ISEED)
C
C  Compute the location of the leading or trailing edge (whichever is closer)
C    of a mechanical chopper at time T, if the half-open/half closed times are
C    TOPEN and TCLOSE, the linear velocity is V (positive if clockwise,
C    negative if counter-clockwise), and the random phase jitter is SIGT.
C    If LOPENING is returned as .TRUE., the result is the location of the
C    opening edge of the chopper; if .FALSE., the closing edge.
C
C  P. A. Seeger, March 15, 1993.
C
C  Variables in calling sequence:
C    T = time that particle reaches chopper (us) (input)
C    TOPEN = nominal time that chopper is half open (us) (input)
C    TCLOSE = nominal time that chopper is half closed (us) (input)
C    SIGT = standard deviation of phase jitter of chopper (us) (input)
C    V = linear velocity of chopper (m/us) (input)
```

```
C    PERIOD = time between chopper openings (us) (input)
C    LOPENING = .true. if chopper is opening, .false. if closing (output)
C    ISEED = random number seed (input/output)
C
      IMPLICIT NONE
      REAL*4 T, TOPEN, TCLOSE, SIGT, V, PERIOD
      LOGICAL LOPENING
      INTEGER*4 ISEED
C
C  Externals:
C    PLNORM
      REAL*4 PLNORM
C--
```