LA-UR-97- .1903

CONF-960935--3

TITLE: THE MCLIB LIBRARY: NEW FEATURES

AUTHOR(S): PHILIP A. SEEGER, LANSCE 12

SUBMITTED TO: PROCEEDING OF WORKSHOP ON METHODS FOR NEUTRON SCATTERING INSTRUMENT DESIGN, BERKELEY, CA SEPTEMBER 23-25, 1996

Los Alamos
NATIONAL LABORATORY

THE MCLIB LIBRARY: NEW FEATURES

Philip A. Seeger
Manuel Lujan Jr. Neutron Scattering Center and Sumner Associates
Los Alamos National Laboratory
Los Alamos, NM 87545 USA

Abstract

This report describes the philosophy and structure of MCLIB, a Fortran library of Monte Carlo subroutines which has been developed to test designs of neutron scattering instruments. Emphasis is placed on new features added to the library since the previous presentation of MCLIB at ICANS-XIII in October, 1995 [1]. These new features include toroidal mirrors, writing and reading source files, splitting and banking of histories, and a Maxwellian probability distribution. The only change of a program structure has been to include charge and polarization vector in the description of a particle. The latest release of the source code and documentation may be obtained by anonymous ftp from *ftp://azoth.lansce.lanl.gov/pub/mclib/*. Work is also continuing on a more friendly web-based user interface, and user input is requested for additional features to be added to the library.

INTRODUCTION

Monte Carlo is a method to integrate over a large number of variables. Random numbers or samples from random distributions are used to select a value for each variable, and the integrand is then evaluated. The process is repeated a large number of times and the resulting values of the integrand 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). Monte Carlo is a useful supplement to analytical treatment of an instrument, in particular to check and demonstrate "non-intuitive" focusing arrangements, but should never be used as a substitute for thinking.

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. To simplify this step, a web page is being developed [2], linked to the home page of the Manuel Lujan Jr. Neutron Scattering Center, *http://www.lansce.lanl.gov/mlnsc/*. This will eventually allow users to design instruments by defining the locations and properties of a variety of beam elements. Whether from a stand-alone program or from this web page, the output is a geometry file containing the complete problem definition.

The geometry file is then passed to a second-stage program, for example MC_RUN (which can be downloaded from *ftp://azoth.lansce.lanl.gov/pub/mclib/* along with the Monte Carlo library and documentation). This will be executed at the user's facility to transport neutrons and tally the 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.

GEOMETRY DESCRIPTION

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. Higher-order surfaces such as toroids which can not be described by Eq. 1 must instead be defined as parameters of a special region (*e.g.*, toroidal mirror, type 14). The scaling of Eq. 1 is arbitrary, but we tend to evaluate non-quadratic surfaces as m (coefficients *B*, *D*, and *F* dimensionless and *G* in m) and quadratic surfaces as m^2 (coefficients *A*, *C*, *E*, *P*, *Q*, and *R* dimensionless, coefficients *B*, *D*, and *F* in m, and *G* in 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.

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; ±5 for cases requiring special action (such as a coordinate transformation) whenever a particle enters or leaves the region; and ±6 if the neutron history is to be split after crossing the surface. Care must be taken to avoid reentrant regions, but provision is made for having a "scattering chamber" region type which may contain embedded regions (*e.g.*, samples and detectors). Surfaces of embedded regions are marked by adding 10 to the surface type number, and 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 also be used with care, since particles within the embedded regions also pass the test for being within the enclosing region.

- Regions have names and have associated types. Defined type numbers are listed here, and the definitions of parameters are given in Appendix A. Types which have been added since the ICANS 13 proceedings [1] are marked "*".

Simple material types:
type 0 = total absorber; no parameters
type 1 = amorphous unpolarized material; 4 parameters
type 2 = aluminum, including Bragg edges; no parameters
type 3 = hydrogenous, including multiple scattering
type 4 = supermirror represented by trapezoidal reflectivity; 4 parameters
type 5 = beryllium at 100K, including Bragg edges; no parameters
type 6 = single-crystal filter, Freund formalism; 3 parameters

Complex regions:
type 10 = multi-aperture collimator
type 11 = multi-slit collimator, vertical blades; 3 or 5 parameters
type 12 = multi-slit collimator, horizontal blades; 3 or 5 parameters
type 13 = crystal monochrometer; 10 parameters
*type 14 = toroidal mirror; 10 parameters

Time-dependent regions:
types 20.n = chopper (disk or blade); 6 parameters
20.0 or 20.2 for motion in x-direction, 20.1 or 20.3 for y-direction
20.2 or 20.3 is counter-rotating (fully closed when edges at 0)
type 21 = Fermi chopper (not yet implemented)
type 22 = gravity focuser; 5 parameters
type 23 = removable beamstop; no parameters

Scattering samples:
type 30 = sample which scatters at constant Q; 2 parameters
type 31 = scattering sample of hard spheres; 2 parameters
type 32 = isotropic scatterer with fixed energy change; 2 parameters
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; $1 + 4N$ parameters
type 36 = scattering from isotropic polycrystalline powder; 6 parameters + $2 \times$ table length

Detectors (zero- and one-dimensional):
type 40 = detector; 9 parameters
type 41 = vertical linear detector; 14 parameters
type 42 = horizontal linear detector; 14 parameters
type 44 = longitudinal linear detector; 14 parameters

Detectors (two-dimensional):
type 43.n = 2-D detector; 19 or 22 parameters

43.0, 43.1, 43.2, rectilinear coordinates, respectively (X,Y), (Z,Y), (X,Z)
43.4, 43.5, 43.6, cylindrical coordinates (rho,phi), respectively about axes parallel to Z-, X-, and Y-axes

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

Sources:
types 90.n = source size and phase space to be sampled; 14-18 parameters
90.1 or 90.2 with rectangular aperture(s)
90.4 with vertical offests of apertures (option numbers may be added)
type 91 = source energy distribution table and line shape parameters; 12 parameters plus length of table
*type 95 = source file, direct-access binary; no parameters, but NAME must be '[path]filename' for the file. The file must be opened with logical unit number 95.

PROGRAM STRUCTURES

The relationships of the structures used by MCLIB are shown schematically in Fig. 1. Each surface is a record of type /SURFACE/, and elements are referenced as *(e.g.)* SURFACE.G. Similarly, a geometric region is a record of type /REGION/, which is a vector *IGEOM* of 2-byte integers, of length equal to the maximum allowed number of surfaces. An additional structure,

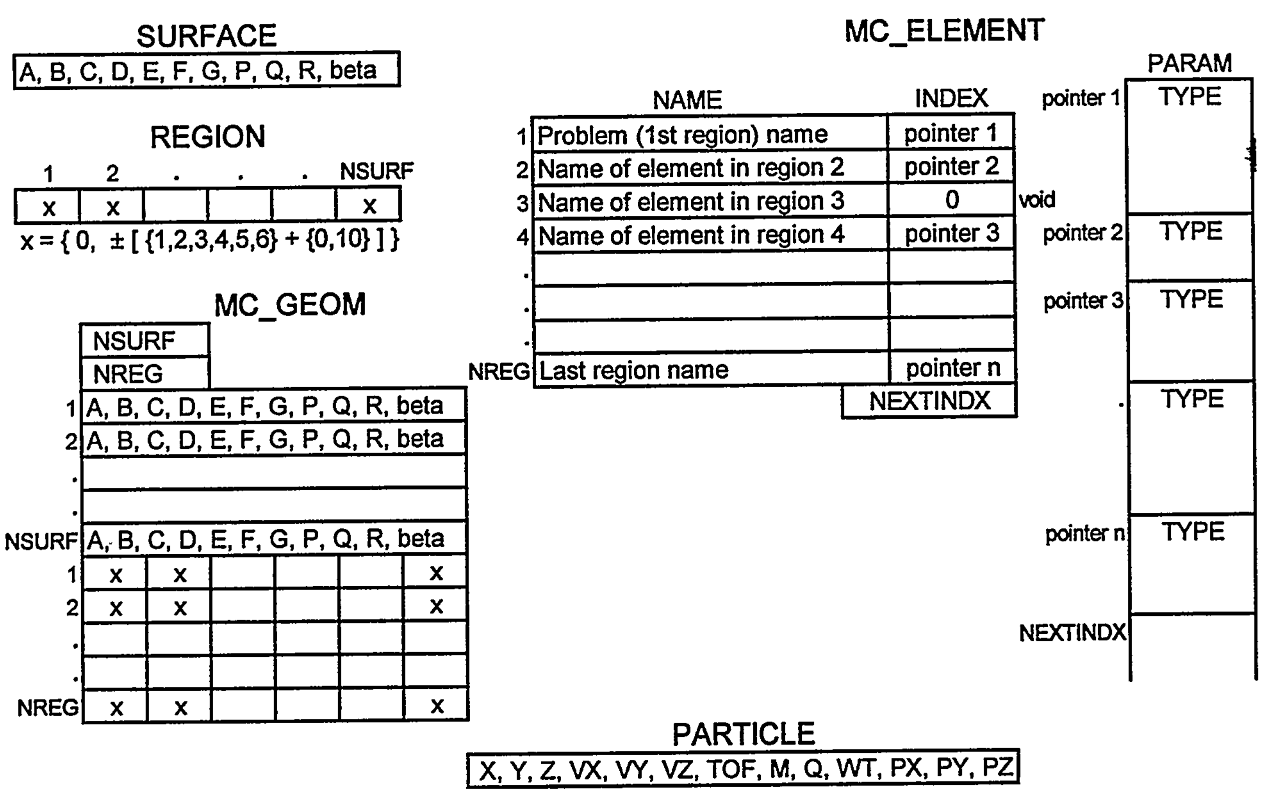

Figure 1. Structures used in MCLIB.

MC_GEOM, contains the numbers of surfaces and regions in the problem, *NSURF* and *NREG*, and arrays of surface and region records. Information about types within the regions 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*. Structures MC_GEOM and MC_ELEMENT represent the complete description of the instrument being simulated.

The final structure, PARTICLE, is the object which is acted upon by the beam element subroutines in the library. It includes position, velocity, time of flight, mass (1 for a neutron), charge (0 for a neutron), statistical weight, and vector polarization (not yet implemented in the code). 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 usually 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.

NEW AND REVISED SUBROUTINES

The following routines have been added or modified. The most up-to-date listing of the subroutine abstracts and of the source codes may be found at *ftp://azoth.lansce.lanl.gov/mclib/*.

ANGTORUS	04 Aug 96	computes the cosine of the angle of incidence with respect to a toroidal surface (including wobble)
DISTORUS	08 Dec 96	finds distance along a trajectory to a horizontal or vertical toroidal shell (including gravity)
GET_SPACE	25 Jan 96	added source brightness to calling sequence
N_SOURCE	25 Jan 96	get source neutron: added Q and polarization vector; may read from direct-access binary file (type 95)
OPERATE	27 Aug 96	find what happens to particle within region: small-angle scattering $\sim\lambda^2$; additional region for chopper blade material; add toroidal mirrors (type 14)
PLMXWLN	25 Mar 96	probability from Maxwellian
PLNORM	23 Jul 96	probability from normal distribution (faster algorithm)
POWDER	07 Dec 95	scatter from polycrystalline powder sample, was always isotropic if $\lambda >$ 2nd edge
SRC_PROB	11 Jul 96	probability density /meV/μs from type 91 table (new)
TESTIN	04 Apr 96	find if within region, now computes in double precision

The program MC_RUN has also had several modifications, most notably:

- prepare direct-access .MON output file for subsequent use as a source, and open and initialize the source file if type=95
- periodic output of data (.DAT) file for protection against computer crashes during long runs
- fix error in refraction into void, including type=50
- bank 20 levels of saved neutrons instead of 1; split by 2 after surface type ±6

- test for "trapped" neutrons which could cause a job to freeze

CREATING A NEW TYPE

The process of adding toroidal mirrors to MCLIB will be used as a case study. There were four steps:

1. Define a new region type (14) and assign parameters and names.
2. Develop needed algorithms and procedures (DISTORUS and ANGTORUS).
3. Insert the "methods" for the new type in subroutine OPERATE.
4. Test, debug, and refine the previous three steps.

1. There is renewed interest in the use of focusing mirrors in neutron instruments, for example the proposal of Benno Schoenborn for a Laue protein-crystallography instrument [3], and a recent paper by John Copley [4]. Although an ellipsoid would be the optically preferred shape, a toroid may be more practical to manufacture. Since a toroid can not be defined as a *surface* by Eq. 1, it was necessary to define it as a *region*. The nature of the region is that it is divided into two subregions by the toroidal surface. The ten parameters can be seen in Appendix A at type 14. The torus is defined by its major radius *R*, the radius *r* of the generating circle, the offset of the center of the generating circle from the beam axis, and the longitudinal position of the center of revolution. Also, the sign of the major radius shows which side of the beam axis the center is on, and the sign of the minor radius is used as a flag for orientation: + for horizontal and – for vertical. (The choice of parameter three as a small offset instead of distance from the center of the torus was crucial to maintaining precision.) The fifth parameter is surface roughness, and then three parameters are provided to define a rotation of the instrument beam axis after reflection. Finally, there are two pointers to define the materials inside and outside the torus. The "inside" material is in front of the mirror surface, and is usually void, indicated by a pointer of zero. The "outside" is the mirror surface/substrate, and the pointer is the offset of its definition in the *PARAM* block relative to the torus parameters; *e.g.*, if the material type immediately follows this block, then the pointer value is 12 (or 1 + *NUMBER_14*, the number of cells in *PARAM* used by type 14).

2. Finding the intersections of a line with a torus (a quartic surface) is not a trivial problem. Previous codes [4,5] have solved the problem very generally with sophisticated methods to retain precision. My approach was to begin with three limiting assumptions and one complication:

- only the near half of the toroid is relevant
- only the outer (concave) half-shell of the toroid is relevant
- only real solutions for the intersections are relevant
- the particle trajectory is a gravitational parabola instead of a straight line.

At this stage I had the assistance of Mike Fitzsimmons and Tom Klugel (MLNSC) to study methods of setting up and solving the fourth-order equation. The first method we used to set up the equation was copied from [5], using analytic derivatives of the trajectory in the coordinate system of the torus. This ran into serious difficulties when gravity was included (see §4 below), and the eventual solution was first to bracket the nearest and furthest intersections (if any!) of the parabola with bounding cylinders and planes of the relevant portion of the torus, and to solve for the distance from the surface parametrically *vs.* time of flight at five equally spaced points. Since the spacing of the points is uniform, multiplication by a predetermined constant matrix solves for the coefficients of the fourth-order equation of nearest distance as a function of time. One or two real roots (if any) are then found by the "safe" Newton-Raphson method discussed in *Numerical Recipes* [6], which uses bisection to keep the roots bracketed whenever the iteration step would move outside the current bracket. For further details, see the source code for DISTORUS in file MCLIB.SRC at *ftp://azoth.lansce.lanl.gov/pub/mclib/*.

Finding the angle of incidence, subroutine ANGTORUS, is far simpler. It is essentially a copy of WOBBLE with derivatives of the torus equation instead of Eq. 1. As in DISTORUS, precision is enhanced by accounting for the fact that the major radius of the torus is large compared to any other distance.

3. Inclusion of type-specific code in subroutine OPERATE requires adherence to the overall structure of the routine, and will probably be the most difficult (or dangerous) aspect for a user to add a function to MCLIB. Here is the scenario:

- When a particle enters a region, OPERATE is called with the following references:
  PART = record containing description of particle (input/output)
  EXDIST = distance to exit surface particle is aimed at (m) (input/output)
  PARAMS = array with description of what is in this region (input)
  GEOM = structure with all surface and region definitions (input)
  IREG = region number of device, or subregion within device (input/output)
  JSURF = surface number, if particle is initially on surface (input)
  KSURF = surface number that particle is pointed toward (input/output)
  NAME = name of region, used as file name for type 34 (input)
  TRANSMIT = flag to compute transmission of sample types 30-39 (input)
  FLAG = flag set to .FALSE. if (*e.g.*) chopper in wrong frame (output)
  PART_2 = description of particle created by operation (output)
  DET_WT = statistical weight of detected particle (output)
  IX, IY = position bin numbers of detected particle (output)
  ISEED = random-number generator seed (input/output)
- OPERATE determines the region type from *PARAMS*(1) and transfers to the appropriate block of code in an IF-THEN-ELSE IF structure (this will become a CASE structure when F90 is more widespread).
- The code may use any variables marked "input" and may change any variables marked "output" in the above list. Local variables will have to be declared, and any variables to be retained across entries must have distinctive names and be placed in a SAVE statement.
- Valid operations include: moving the particle to the exit from the region with possible reeducation of statistical weight to account for absorption (set *EXDIST* to 0); selection of a subregion without moving; reflection (set *KSURF* negative); rotate the coordinate system; statistically split the particle or create a new one (in *PART_2*); scatter either elastically or inelastically; multiple scattering, keeping control through several motions of the particle till it reaches the region exit; detect the particle and find encoded position in detector. Examples of all these possibilities can be found in OPERATE.FOR.
- For compatibility with older Fortran compilers without recursive subroutines, the code should not make any calls to OPERATE for processing subregions. Use no numbered statements. Style should be similar to the existing OPERATE.FOR code.

In the case of the toroidal mirror, the actions are to determine the distance to the torus and which side the particle is on using DISTORUS, and to move either to that surface or the region exit (whichever is closer) through the appropriate material, compute reflection probability using ANGTORUS and the ratio of scattering length densities across the surface, and repeat all of these actions till the region is exited. Note that *only* material types 0 (void), 1 (amorphous non-magnetic), or 4 (supermirror layer) are allowed, due to the restriction that all code for the type must be inline. In fact, the code for type 1 and type 4 regions has been copied into this block.

4. The debugging of this module is an example of cooperation with outside users. The code had been tested by reproducing the results in [4], but in a different coordinate system. When John Copley tried to use the code, there were subtle errors. It took one more test to show that my code was at fault, rather than John's interpretation of it, two more rounds of modification and testing to uncover a serious philosophical flaw in the algorithm, and two more rounds to "perfect" the code. This version passes all tests, and is twenty times faster [7] than the program used previously.

SMALL-ANGLE SCATTERING

The original paper on curved-mirror neutron optics [8] suggested small-angle scattering as an application. The authors considered a bent mirror forming a cylinder with axis transverse to the beam, and a narrow slit aperture with the solid angle of the mirror matched to the solid angle of the neutron source. The same geometrical matching can be applied to illuminate a mirror which focuses in two dimensions, as illustrated in Fig. 2. (Note that the transverse dimensions in Fig. 2 are exaggerated by a factor of 10 relative to the longitudinal dimensions.) The MCLIB code allows us to compare mirror shapes in this geometry: the ideal ellipsoid, the tangent toroid, cylindrical segments (with longitudinal cylinder axes), or toroidal segments (bent cylinders with varying curvatures). Compared to a two-aperture collimation system with the same resolution, a factor of 10 increase in intensity is possible because the full illuminated surface of the moderator may be viewed. The tradeoff between intensity and resolution depends on the single aperture.

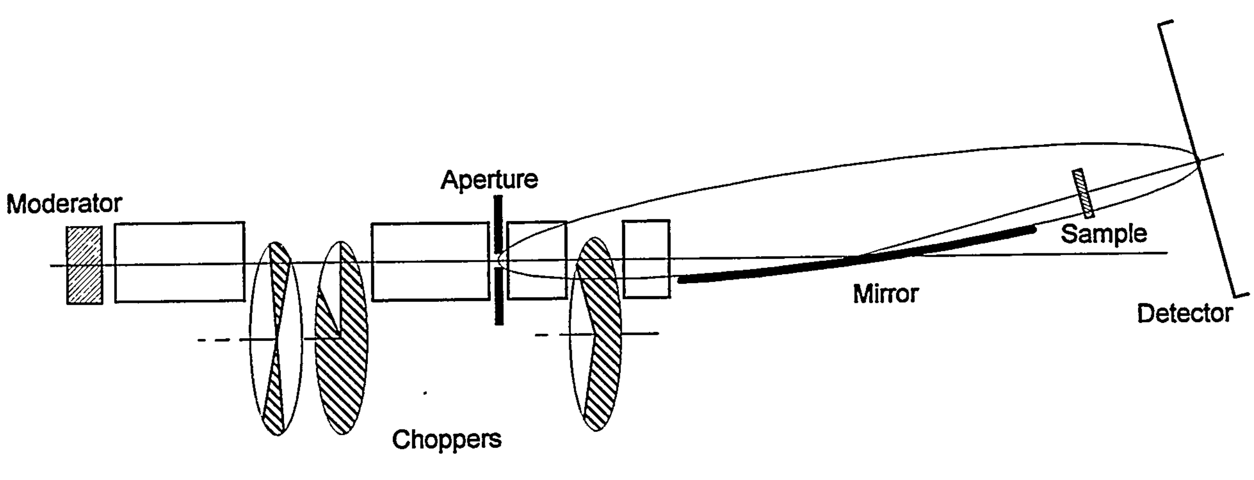

Figure 2. Small-Angle Scattering instrument using a focusing mirror for increased intensity.

The mirror considered as an example was 3.0 m long and 75 mm wide, with a major axis of 9 m and transverse (minor) axes of 157 mm. The glancing is 1° at the center (beam axis bent by 2°). The radius-of-curvature parameters of the tangent toroid are R = 257.77 m and r = 78.54 mm. The source aperture had a diameter of 2.0 mm, located at one focus of the ellipsoid; at this eccentricity, the focus is less than 1 mm from the vertex. Also, the transverse diameter of the ellipsoid at the focus is only 2.7 mm, so the source is "large" in terms of optical properties, and Monte Carlo is a useful tool a estimating resolution at the detector placed at the other focus. The detector pixel size was made small (0.25 mm) and the encoding uncertainty was zero so as not to affect the computed resolution when compared to the source aperture size, which contributes 0.50 mm to the standard deviation in each detector coordinate. Since the effect of gravity is always included in the algorithms of MCLIB (for any particles with rest mass), it is necessary to place the source aperture (and the detector center) below the instrument axis *if the reflection plane is horizontal* by a distance which is proportional to wavelength, such that the nominal trajectory at the mirror center is horizontal. However, *if the reflection is upwards* (mirror horizontal and concave up), then no correction is necessary because the excess downward velocity acquired before striking the mirror is reflected and then canceled by downward acceleration after reflection (this cancellation is exact only for neutrons reflected at the center of the mirror, but is independent of wavelength). For a white source (*i.e.,* spallation) the vertical reflection plane is much to be preferred.

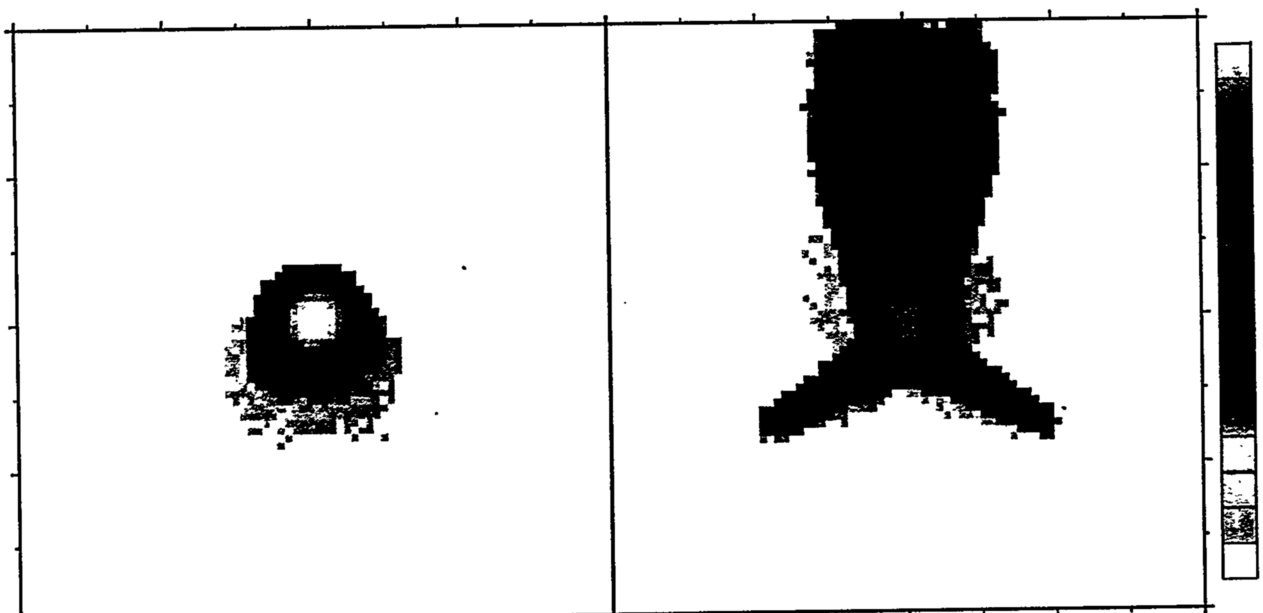

Figure 3. Images of a 2-mm diameter aperture for ellipsoidal (left side) and toroidal (right side) focusing mirrors. The reflection plane is vertical (upward) to correct for gravity. Each pattern is a histogram for 250000 detected neutrons with wavelength distribution from 2–20 Å, from a coupled liquid-hydrogen moderator. Each figure is 20 mm square on the detector (printed at four times size), and the intensity scale is logarithmic from 1 to 93250 counts per pixel.

A comparison of the beam spots for the ellipsoid and toroid is shown in Fig. 3, for vertical reflection of a cold-moderator spectrum from 2–20 Å. (A supermirror reflecting layer was assumed, and reflectivity is good above 3 Å.) The standard deviations for the ellipsoid are 0.64 mm in both the horizontal and vertical directions, and those for the toroid are 1.05 mm horizontal and 2.29 mm vertical. A minimum Q-value less than 0.001 $Å^{-1}$ could be achieved with this geometry in an instrument of total length 13 m (with the ellipsoidal mirror). Further details of this simulation will be presented later.

REQUESTS FOR ADDITIONAL ELEMENT TYPES

It is hoped that the MCLIB code will be generally useful to designers of neutron instruments. For this goal to be realized, two sets of improvements are being pursued. First, the general user interface is being moved to a web page so that geometry files may be constructed more easily than is now possible. Second, we are soliciting input of algorithms from interested or potential users, so that the code will do what *you* want. Please communicate with the author by e-mail to *PASeeger@aol.com* if you have any questions or suggestions.

ACKNOWLEDGEMENT

This work was supported by the U. S. Department of Energy, Office of Basic Energy Sciences and other Department of Energy programs under Contract W-7405-ENG-32 to the University of California.

REFERENCES

1. P. A. Seeger, "The MCLIB Library: Monte Carlo simulation of neutron scattering instruments," 13th Meeting of the International Collaboration on Advanced Neutron Sources, October 11–14, 1995, Paul Scherrer Institut, Villigen, Switzerland, *PSI Proceedings* **95-02**, pp. 194–212.

2. T. G. Thelliez, L. L. Daemen, P. A. Seeger, and R. P. Hjelm, Jr., "A user-friendly geometry interface for the Monte Carlo neutron optics code MCLIB," 13th Meeting of the International Collaboration on Advanced Neutron Sources, October 11–14, 1995, Paul Scherrer Institut, Villigen, Switzerland, *PSI Proceedings* **95-02**, pp. 307–311.

3. B. P. Schoenborn and E. Pitcher, in *Neutrons in Biology,* B. P. Schoenborn and R. B. Knott, eds., Plenum Press, New York, pp. 433–444 (1996).

4. J. R. D. Copley, *Rev. Sci. Instrum.* **67** (1996) 188–194.

5. G. Hummer, subroutine TORUS, Feb. 28, 1995.

6. W. H. Press, S. A. Teukolsky, W. T. Vettering, and B. P. Flannery, *Numerical Recipes in Fortran, Second Ed.*, Cambridge University Press, New York, §9.4 (1992).

7. C. Lartigue, private communication.

8. H. Maier-Leibnitz and T. Springer, *J. Nucl. Energy Parts A/B* **17** (1963) 217–225.

APPENDIX A. 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                 (reflectrometry) [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    16 Oct 1995: changed parametrization of supermirror (type 4) [PAS]
C    11 Nov 1995: add NUMBER_nn to all types [PAS]
C    22 Nov 1995: add subtypes and coordinate origin to type 43 [PAS]
C    24 Jan 1996: type 95, source from direct-access binary file [PAS]
C    05 Aug 1996: type 14, toroidal mirror [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
           INTEGER NUMBER_00
           PARAMETER (NUMBER_00=1)
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, NUMBER_01
           PARAMETER (REAL_RHO=1,IMAG_RHO=2,NSIGMA0=3,NSIGMAV=4,
     1                NUMBER_01=5)
C  type 2 = aluminum, including Bragg edges; no additional parameters
           INTEGER NUMBER_02
           PARAMETER (NUMBER_02=1)
C  type 3 = hydrogenous, including multiple scattering; 1 parameter
C          Relative hydrogen density compared to water
           INTEGER H_DENSITY, NUMBER_03
           PARAMETER (H_DENSITY=1,NUMBER_03=2)
C  type 4 = supermirror represented by trapezoidal reflectivity; 4 parameters
C          Real & Imaginary scattering-length density (10**10/cm**2) (see type 1)
C          Supermirror multiplier
C          Refectivity at maximum supermirror limit
```

```
      INTEGER SUPER_MULT, SUPER_REFL, NUMBER_04
      PARAMETER (SUPER_MULT=3,SUPER_REFL=4,NUMBER_04=5)
C  type 5 = beryllium at 100K, including Bragg edges; no additional parameters
      INTEGER NUMBER_05
      PARAMETER (NUMBER_05=1)
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, NUMBER_06
      PARAMETER (XSIGFREE=1,X_C2=2,XSIGABS=3,NUMBER_06=4)
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, NUMBER_11
      PARAMETER (B_SIN_PHI=4,B_COS_PHI=5,NUMBER_11=6)
C  type 12 = multi-slit collimator, horizontal blades; 5 parameters
C          Same parameters as type 11
      INTEGER NUMBER_12
      PARAMETER (NUMBER_12=6)
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, NUMBER_13
      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,NUMBER_13=11)
C  type 14 = toroidal mirror; 10 parameters
C       Radius of rotation of torus, from axis to center of generating
C         circle; positive if torus axis right of or above beam axis (m)
C       Radius of cross section of torus; negative if torus vertical (m)
C       Offset of center of generating circle from beam axis (m)
C       Z-coordinate of torus axis (m)
C       Surface roughness parameter
      INTEGER TORUS_A, TORUS_B, TORUS_D, TORUS_Z, TOR_BETA
C       sin, cos, and Z-center for beam axis rotation (if any)
      INTEGER TOR_SIN_TH, TOR_COS_TH, TOR_Z_ROT
C       Offsets to parameter blocks for interior and exterior regions,
```

```
C          zero if void; types 1 and 4 are supported
        INTEGER INSIDE_OFFSET, OUTSIDE_OFFSET, NUMBER_14
        PARAMETER (TORUS_A=1, TORUS_B=2, TORUS_D=3, TORUS_Z=4,
     1              TOR_BETA=5, TOR_SIN_TH=6, TOR_COS_TH=7, TOR_Z_ROT=8,
     2              INSIDE_OFFSET=9, OUTSIDE_OFFSET=10, NUMBER_14=11)
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, NUMBER_20
        PARAMETER (CHP_VEL=1,CHP_HALF=2,CHP_OPEN=3,CHP_CLOSE=4,
     1             CHP_JITTER=5,CHP_PERIOD=6,NUMBER_20=7)
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, NUMBER_22
        PARAMETER (G_ACCEL=1,G_JITTER=2,G_START=3,G_TOP=4,G_PERIOD=5,
     1             NUMBER_22=6)
C  type 23 = removable beamstop; no additional parameters
        INTEGER NUMBER_23
        PARAMETER (NUMBER_23=1)
C
C       first parameter of many 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, NUMBER_30
           PARAMETER (Q_SCATTER=2,NUMBER_30=3)
C  type 31 = scattering sample of hard spheres; 1 additional parameter
C          hard-sphere radius for scatter (A)
           INTEGER R_SPHERE, NUMBER_31
           PARAMETER (R_SPHERE=2,NUMBER_31=3)
C  type 32 = isotropic scatterer with constant energy change; 1 additional
C          inelastic energy change (0 if elastic) (meV)
           INTEGER DELTA_E, NUMBER_32
           PARAMETER (DELTA_E=2,NUMBER_32=3)
C  type 34 = inelastic scattering using MCNP file; no parameters, but NAME
C       must be '[path]filename' for the file
        INTEGER NUMBER_34
        PARAMETER (NUMBER_34=1)
C  type 35 = scattering from layered reflectrometry 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,
     1          NUMBER_35
        PARAMETER (NLAYERS=1, REAL4PINB=2, IMAG4PINB=3,
     1            THK_LAYER=4, ROUGHNESS=5, NUMBER_35=2)
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,
     1          NUMBER_36
        PARAMETER (N_BRAGG=1,PSIGMAT=2,PSIGMAI=3,PSIGMAS=4,PSIGMAA=5,
     1            D_BRAGG=6,NUMBER_36=7)
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         electronic delay of detector events (us)
C         repeat period of data-acquisition electronics (us)
C       Note: if t-o-f is logarithmic, TMAX is overriden
        INTEGER D_TMIN, D_TMAX, D_TCHANS, D_DT_OVER_T, D_TICK, D_DELAY,
     1          D_T_PERIOD, NUMBER_40
        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,NUMBER_40=10)
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,
     1            NUMBER_41
          PARAMETER (DET_YMIN=10,DET_YMAX=11,DET_NY=12,DET_DELY=13,
     1              DET_RMSY=14,NUMBER_41=15)
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,
     1            NUMBER_42
          PARAMETER (DET_XMIN=10,DET_XMAX=11,DET_NX=12,DET_DELX=13,
     1              DET_RMSX=14,NUMBER_42=15)
```

```
C  type 43.n = 2-D detector; 19 or 22 additional parameters
C        .0,.1,.2, rectilinear coordinates, respectively (X,Y), (Z,Y), (X,Z)
C        .4,.5,.6, cylindrical coordinates (rho,phi), respectively about
C                  axes parallel to Z-, X-, and Y-axes
C          locations of "left" and "right" edges of detector (m)
C          number of detector elements in the X (or Z or rho) 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 or rad)
C          number of detector elements in the Y (or Z or phi) direction
C          height of detector element (m or rad)
C          root-mean-square Y encoding error of detector (m)
C          origin of cylindrical coordinates (subtypes .4,.5,.6) (m)
           INTEGER DET2_XMIN, DET2_XMAX, DET2_NX, DET2_DELX, DET2_RMSX
           INTEGER DET2_YMIN, DET2_YMAX, DET2_NY, DET2_DELY, DET2_RMSY
           INTEGER DET2_X0, DET2_Y0, DET2_Z0, NUMBER_43
           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,DET2_X0=20,DET2_Y0=21,
     4                DET2_Z0=22,NUMBER_43=23)
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,
     1             NUMBER_44
           PARAMETER (DET_ZMIN=10,DET_ZMAX=11,DET_NZ=12,DET_DELZ=13,
     1                DET_RMSZ=14,NUMBER_44=15)
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.
         INTEGER NUMBER_50
         PARAMETER (NUMBER_50=1)
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, NUMBER_90
      PARAMETER (APTR1_Y_OFFSET=17,APTR2_Y_OFFSET=18,NUMBER_90=19)
C  type 91 = source energy distribution table and lineshape parameters; 14
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         or mean (t/lambda) parameter of Ikeda-Carpenter (us/A) (if width<0)
C       Gaussian delay and width, /lambda (us/A) (width < 0 is special case)
      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, NUMBER_91
      PARAMETER (E_TABLE=13,NUMBER_91=13)
C  type 95 = source file, direct-access binary; no parameters, but NAME must
C       be '[path]filename' for the file.  File format is
C       1st record: 16 bytes of coding information and 17-character ID
C       2nd record: 40-character TITLE of job that created the file
C       3rd record: source surface definition
C       4th record: # of histories (integer*4), sum of weights, source MW-s,
C                   modified source brightness, phase space, lethargy
C       records 5-(# of histories+4): source neutrons
      INTEGER NUMBER_95
      PARAMETER (NUMBER_95=1)
C
```