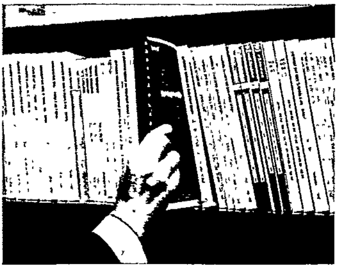
N68-13509

SYNTHESIS OF CALCULATIONAL METHODS FOR THE DESIGN AND ANALYSIS OF RADIATION SHIELDS FOR NUCLEAR ROCKET SYSTEMS. VOLUME 9: FASTER - A FORTRAN ANALYTIC SOLUTION OF THF TRANSPORT EQUATION BY RANDOM SAMPLING

T. M. Jordar



REPORT

selection aids

Pinpointing R & D reports for industry

Clearinghouse, Springfield, Va. 22151

U.S. GOVERNMENT RESEARCH AND DEVELOPMENT REPORTS (USGRDR)---SEMI-MONTHLY JOURNAL ANNOUNCING R&D REPORTS. ANNUAL SUBSCRIPTION $30.00 ($37.50 FOREIGN MAILING). SINGLE COPY $3.00.

U.S. GOVERNMENT RESEARCH AND DEVELOPMENT REPORTS INDEX---SEMI-MONTHLY INDEX TO U.S. GOVERNMENT RESEARCH AND DEVELOPMENT REPORTS. ANNUAL SUBSCRIPTION $22.00 ($27.50 FOREIGN MAILING). SINGLE COPY $3.00.

FAST ANNOUNCEMENT SERVICE---SUMMARIES OF SELECTED R&D REPORTS COMPILED AND MAILED BY SUBJECT CATEGORIES. ANNUAL SUBSCRIPTION $5.00, TWO YEARS: $9.00, AND THREE YEARS: $12.00. WRITE FOR AN APPLICATION FORM.

DOCUMENT PRICES---ALMOST ALL OF THE DOCUMENTS IN THE CLEARINGHOUSE COLLECTION ARE PRICED AT $3.00 FOR PAPER COPIES AND 65 CENTS FOR COPIES IN MICROFICHE.

COUPONS---THE CLEARINGHOUSE PREPAID DOCUMENT COUPON SALES SYSTEM FOR PURCHASING PAPER COPIES AND MICROFICHE PROVIDES FASTER, MORE EFFICIENT SERVICE ON DOCUMENT REQUESTS. THE PREPAID COUPON IS A TABULATING CARD WITH A FACE VALUE OF THE PURCHASE PRICE OF A CLEARINGHOUSE DOCUMENT ($3.00 PAPER COPY OR 65 CENTS MICROFICHE). IT IS YOUR METHOD OF PAYMENT, ORDER FORM, SHIPPING LABEL, AND RECEIPT OF SALE.

COUPONS FOR PAPER COPY (HC) DOCUMENTS ARE AVAILABLE AT $3.00 EACH OR IN BOOKS OF 10 COUPONS FOR $30.00. COUPONS FOR MICROFICHE COPIES OF CLEARINGHOUSE DOCUMENTS ARE AVAILABLE IN BOOKS OF 50 COUPONS FOR $32.50. WRITE FOR A COUPON ORDER FORM.

WANL-PR(LL)-010
June, 1967

Westinghouse Astronuclear Laboratory

SYNTHESIS OF CALCULATIONAL METHODS FOR THE DESIGN AND ANALYSIS OF RADIATION SHIELDS FOR NUCLEAR ROCKET SYSTEMS

Contract No. NAS-8-20414
Contract No. DCN-16-28-0029(IF)

FINAL PROGRESS REPORT

Volume 9

Astronuclear Laboratory

WANL-PR-(LL)-010

VOLUME 9

SYNTHESIS OF CALCULATIONAL METHODS FOR THE DESIGN AND ANALYSIS OF RADIATION SHIELDS FOR NUCLEAR ROCKET SYSTEMS

FASTER

A FORTRAN ANALYTIC SOLUTION OF THE TRANSPORT EQUATION BY RANDOM SAMPLING

by

T. M. Jordan

Contract No. NAS-8-20414
Contract No. DCN-16-28-0029(IF)

ACKNOWLEDGMENT

The authors appreciate the guidance provided by Mr. Henry E. Stern, Deputy Manager, Nuclear and Plasma Physics Division, George C. Marshall Space Flight Center, the technical monitor of the contract.

Astronuclear
Laboratory

ABSTRACT

This report is Volume 9 of nine volumes of the final report on "Synthesis of Calculational Methods for the Design and Analysis of Radiation Shields for Nuclear Rocket Systems". Presented in this volume is a description of the FASTER program, a Fortran Analytic Solution of the Transport Equation by Random sampling.

FASTER is a Fortran IV Monte Carlo program which calculates energy dependent neutron or photon fluxes at points, surfaces and regions of complex geometries. This program contains all the data processing routines required for a wide variety of nuclear vehicle applications. The FASTER program is completely variable dimensioned, and hence, is capable of treating problems of varying complexity within the limitation of total computer storage. The program uses only the input and output tape units and is operational on the 32K WANL and MSFC IBM 7094 computers--using overlay--and on the 64K CDC 6600 computer at the Westinghouse Telecomputer Center in Pittsburgh.

FASTER utilizes the general quadric surface equation for describing the geometry. The more common equations for planes, cones, elliptical cylinders and ellipsoids can also be used for input description of the surfaces.

The FASTER program handles either neutron or photon sources. Each source is described in rectangular, cylindrical or spherical coordinates and the source geometry is superimposed on the problem geometry. The spatial, angular, and energy source distributions are assumed to be separable and are input as tabulated relative distribution data.

The FASTER program deals with the entire spectrum of particle energies simultaneously, thus eliminating costly repetition of geometric calculations which are usually required for treating individual mono-energetic particles. The scattered particle energy spectra include the effect of every possible scattering event at each scattering point. This eliminates the variance associated with the random selection of a single event. Neutron transport calculations utilize averaged multigroup cross sections which are available from several standard tabulations. Photon transport problems utilize point cross sections and the Klein-Nishina equation for Compton scattering.

Biased random sampling is used exclusively in the selection of source and scattering points. In performing flux calculations at a point, the FASTER program includes the singularity of the point flux estimator in the biasing functions. This permits more economical and more accurate point solutions in source and/or scattering volumes.

Analytic estimation is used to compute point fluxes and/or surface and volume averaged fluxes. These fluxes can be processed within the program to obtain various responses such as dose rates and heating rates. Contributions to the fluxes are obtained by source, number of collisions, and scattering region. Legendre angular moments and length-of-flight moments of the fluxes and responses can also be obtained.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1.0 INTRODUCTION AND SUMMARY

A major problem in the analysis of nuclear rockets is accuracy in predicting neutron and photon radiation levels both internal and external to the radiation sources. The most useful and fundamental expression of these radiation levels consists of energy dependent fluxes at specified points. These point fluxes can be utilized with a variety of energy dependent functions to yield integral quantities such as dose rate and heating rate. These point quantities can also be spatially integrated to obtain averages or totals for various surfaces and volumes, e.g., neutron and photon reactor leakage and volumetric heating. The accurate calculation of point fluxes requires a detailed treatment of source and material distributions and of basic particle cross sections. Accurate and economical calculations are particularly difficult for space vehicles and nuclear rocket test stands because of their complex geometries. The FASTER program was developed to handle these geometrically complex problems.

Energy dependent fluxes can be obtained, in principle, for any radiation analysis problem by computing the flux contributions by order of scatter. The geometric complexity of most realistic problems, however, limits a conventional numerical integration to the uncollided and, at most, the single scattered flux components. Therefore, the method of random sampling, i.e., the Monte Carlo method is used in the FASTER program. The Monte Carlo method permits a calculation of the flux components to an arbitrary order of scatter while simplifying the numerical procedures.

The FASTER program is an integral part of the "final" design method schematically shown in Fig. 1. This "final" design method is described more fully in Volume 1 of this report. As depicted in the figure, it begins with the POINT program (Volume 2) which prepares cross section and other basic data for use in the ODD-K two-dimensional transport program. The ODD-K two-dimensional transport program (Volume 6) provides neutron and photon energy fluxes throughout the reactor geometry. The NAGS data processing program (Volume 7) processes these fluxes and calculates neutron and photon radiation levels and neutron and photon source distributions within the reactor system. These sources can be employed in either the KAP-V point kernel program (Volume 4) or the FASTER Monte Carlo program for obtaining

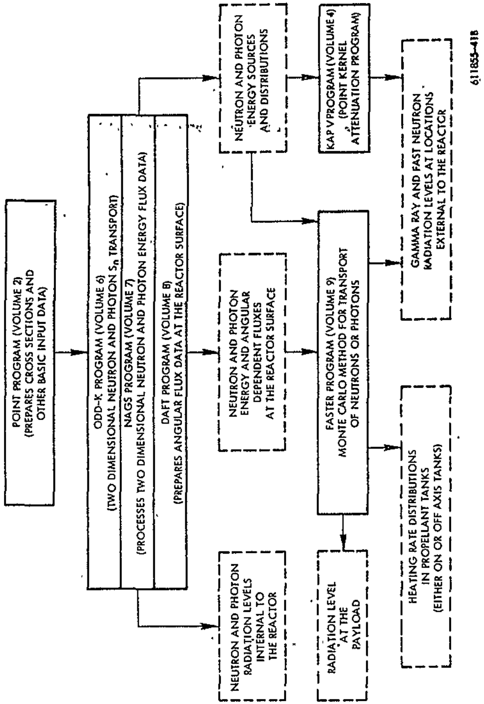


Figure 1. Schematic Diagram of the "Final" Design Method

Astronuclear Laboratory

radiation levels at locations internal and external to the reactor system. In addition, the FASTER program can compute heating rate distributions in the liquid hydrogen propellant (in either an on-axis or off-axis tank) and the radiation level at the payload. Alternately, the DAFT program (Volume 8) can prepare neutron and photon energy and angular dependent fluxes at the reactor surface from the ODD-K program for use in the FASTER Monte Carlo code.

Distinctive features of the Monte Carlo method as employed in the FASTER program, are described in Section 2.0 and include:

1) application of random sampling to the spatial and angular integrations only,
2) consistent use of energy-averaged sampling functions,
3) approximation of importance functions by point kernel techniques,
4) analytic treatment of the energy variable over its entire range, and
5) zero variance energy integration of the scattered source equations.

The Monte Carlo method presented in this report is based on techniques described in References 1, 2, and 3. Considerable improvement in the treatment of point detectors has been made since these early efforts were reported. In particular, the singularity in the point detector flux estimator has been included in the spatial sampling functions without sacrificing a detailed treatment of geometric effects.

The FASTER program utilizes this improvement of the Monte Carlo method to perform neutron or photon transport calculations in complex geometries. Computer oriented features of this program include:

1) coded in FORTRAN IV,
2) completely variable dimensioned,
3) completely internal--auxiliary tapes are not required, and
4) compatible with both the IBM 7094 and CDC 6600 computers.

The program logic is discussed in Section 3.0.

Subsequent sections of this report deal with the numerical techniques used in the FASTER program to implement the Monte Carlo method. First, is the geometric framework which utilizes the quadric surface equation. Basic features of the geometry described in Section 4.0 include:

1) separate description of surfaces,

2) simple input for the more common planes, cones, elliptical cylinders, and ellipsoids with an internal expansion to obtain the coefficients of the general quadric equation,

3) exclusive use of the general surface equation in all computations,

4) region description by listing bounding surfaces,

5) an internal calculation of the (±) sign associated with region boundaries by using the coordinates of an arbitrary point in each region,

6) geometry consistency check using "point-in-region",

7) an internal calculation of "most-probable-next-regions" for boundary crossings,

8) an internal calculation of exterior boundaries, and

9) elimination of the (ϵ, δ) boundary crossing search.

The FASTER program will treat multiple fixed sources where each source has separable spatial, angular and energy distributions. Other features of the fixed source description, described in Section 5.0, include:

1) rectangular, cylindrical and/or spherical coordinates,

2) each spatial variable can be continuous or discrete, permitting a variety of point, line, surface, and volume source geometries,

3) each angular variable can be continuous or discrete, running the range from monodirectional to angular,

4) spatial and angular distributions are specified by tabulating relative distributions at discrete points, in particular the distributions calculated by the NAGS and DAFT programs,

5) spectra are specified by tabulating either differential number or energy spectrum, or groupwise integrated number or energy spectrum, either one in an arbitrary group structure,

6) each source is normalized to total energy or particles.

Features of the treatment of cross sections for photons and neutrons are described in Section 6.0 and include:

1) compositions are accepted in 10^{24} atoms/cm^3 or gm/cm^3,

2) microscopic cross sections can be in barns/atom or cm^2/gm,

3) photon total cross sections are defined at group boundaries and interpolated linearly in energy,

4) photon scattering uses the Klein-Nishina equation,

5) photon energy absorption coefficients are computed internally by element and composite material,

6) group averaged neutron cross sections are accepted from several standard tabulations, e.g., GAM-1,

7) neutron scattering cross sections are not limited in down scatter or order of the Legendre angular expansion,

8) neutron cross sections are manipulated internally to define transport corrected values or to remove this correction,

9) neutron kinetic heating responses are computed by element and composite material, and

10) hydrogen densities can be specified by region, eliminating the need of describing several composite materials which differ by hydrogen content only.

FASTER computes various flux moments in a collapsed set of energy groups for point, surface and/or volume detectors. All quantities are obtained by "analytic estimation", as shown in Section 7.0, and include:

1) groupwise number and energy flux, average energy, and variance,

2) groupwise differential and cumulative number and energy fluxes,

3) groupwise responses and total responses with limits on variance,

4) groupwise number flux and response function totals by source,

5) groupwise flux and response totals by scattering region,

6) groupwise flux and response totals by number-of-collisions,

7) Legendre angular moments of the groupwise flux and response totals, and

8) length-of-flight moments of the groupwise flux and response totals.

The sampling functions incorporated in FASTER, described in Section 8.0, are relatively easy to use. Typical random sampling input data or biasing data for point flux calculations consist of:

1) center and radius of a sphere enclosing all fixed sources,
2) group importance, e.g., flux-to-dose response,
3) linear buildup coefficient by group,
4) relative importance by group of forward-to-backward scattering for heavy elements,
5) relative importance by group of forward-to-backward scattering for hydrogen, and
6) a set of adjustment factors (ratios ≈ 1.0) applied to internally calculated sampling parameters.

Detailed input instructions are presented in Section 9.0 of this report. Section 10.0 contains a description of the input and output for a sample problem involving the calculation of photon and neutron fluxes at a point above a liquid hydrogen propellant tank. The FORTRAN IV listing of the FASTER program is given in Appendix C.

Computer times for the FASTER program depend on the problem complexity and the manner in which the program is used. Individual point detector flux calculations in and around a nuclear reactor require about 3 minutes per point for photons (23 groups) and 6 minutes per point for fast neutrons (13 groups). These times are typical for the IBM 7094 computer and yield computed variances in integral responses generally less than 10 percent.

Flux calculations for points in void regions and/or flux calculations for surfaces and volumes require more computer time per problem, but the fluxes are all obtained simultaneously. A typical photon problem involving volumetric heating rates for 50 regions of a reactor required about 1.2 hours on the IBM 7094 with computed variances generally less than 25 percent.

SECTION 2.0 THE MONTE CARLO METHOD

This section describes the Monte Carlo Method as used in the FASTER program. The first topic is a general discussion of order-of-scatter solutions of the transport equation. The order-of-scatter equations are written explicitly starting with the equation for the uncollided scalar flux. The basic concepts of random sampling are then summarized and applied directly to the order-of-scatter equations. This is followed by a discussion of point kernel estimates of optimal sampling functions. Finally, the equations for angular flux estimation are derived for point, surface, and volume detectors.

2.1 GENERAL COMMENTS

Fluxes at a point can be obtained in principle for any geometry by computing the order-of-scatter components, i.e., the uncollided flux, the single scattered flux, etc. This method utilizes a known fixed source distribution, which is numerically integrated with a simple attenuation kernel over the spatial extent of the source, to obtain the uncollided angular flux at all points in space. This uncollided flux is integrated with the scattering cross sections, over energy and solid angle, to yield the single scattered source distribution. Next, the procedure is repeated to an arbitrary order of scattering, finally yielding the total flux within an inherent error given by the uncomputed higher order-of-scattering flux components.

The FASTER program utilizes the Monte Carlo method in performing the numerical integrations. The Monte Carlo method involves an application of random sampling to the evaluation of definite integrals and it can be applied to all of the integrations involved in the order-of-scatter solution. Because this method is statistical in nature its application to each integration must be accompanied by an iteration procedure to reduce the associated variance. Therefore, it is desirable to perform as many of the integrations as possible by conventional techniques. For the order-of-scatter solution utilized in the FASTER program, the Monte Carlo method is applied only to the spatial and angular integrations, i.e., the method is applied to the variables that complicate direct numerical integration.

The application of the Monte Carlo method can proceed in many directions. The procedure used in the FASTER program involves the random selection of a single point characterizing the distributed fixed source. Then, the fixed source is evaluated at this point and the resulting point source is used to compute energy dependent uncollided fluxes throughout the geometry. Next, a first-scatter point is selected by random sampling. An energy-dependent angular flux (monodirectional) at this first scatter point is computed using the point representation of the fixed source. This energy dependent angular flux, when integrated with the scattering cross sections, yields a point representation of the single-scattered source for all scattered energies and directions. This single scattered point source is used to obtain the energy dependent single-scattered fluxes throughout the geometry. It is used to calculate the monodirectional flux at the next scattering point (obtained by random sampling). This procedure is continued to an arbitrary order-of-scatter and is repeated (starting with the fixed source) until the statistical error in computed fluxes is acceptable.

An implicit difficulty with this point-to-point procedure is the singularity in the attenuation kernel resulting from the inverse square law. This difficulty is present in the calculation of fluxes at both the next scatter point and at arbitrary detector points. It shall be seen later, that these difficulties are removed by considering the singularities in the random selection of the discrete position vectors.

An important consideration in every numerical integration is the selection of a procedure which will yield minimum error. Since, the Monte Carlo method is just another way of performing the spatial integrations, the same considerations are present. It is possible to devise many techniques for reducing the error in an integration performed by random sampling. The most fruitful technique involves the concept of spatial importance. Thus, in considering each possible source or scattering point, the following question is asked: How important will the selection of this point be to the final answer, i.e., how much flux, dose, etc. will eventually be derived from its selection?

Without solving the problem, there is one method which is readily available for approximating spatial importance. This is the point kernel method which approximates the

importance of future scattering events by simple alterations of the attenuation kernel, i.e., removal theory for neutrons and buildup factors for photons. This procedure will not yield a zero variance (or error) flux calculation, but it will yield information as to the approximate importance of space points. This importance function includes the singularities already mentioned and also approximates the importance of material and fixed source distributions.

2.2 THE ORDER-OF-SCATTER EQUATIONS

The solution of the time independent transport equation can be obtained by computing the order-of-scatter components of the flux. This method utilizes a known, differential source density where the uncollided flux is given by the familiar equation:

$$\phi_o(\vec{r}, E) = \iiint S_o(\vec{r}\,', \vec{\Omega}, E) \, \frac{\exp\left[-\int_o^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right]}{s^2} \, dV \qquad (2.1)$$

where

$s = |\vec{r} - \vec{r}\,'|$, $\quad \vec{\Omega} = (\vec{r} - \vec{r}\,')/S$

$\vec{r}, \vec{r}\,'$ are position vectors

$\vec{\Omega}$ is a unit direction vector

E is the particle energy (Mev)

s, s' are scalar distances from $\vec{r}$ along $-\vec{\Omega}$ (cm)

dV is a differential volume element (cm^3)

$S_o(\vec{r}, \vec{\Omega}, E)$ is the fixed differential source density $\left(\frac{\text{particles}}{cm^3 \text{ steradian Mev} \cdot \text{sec}}\right)$

$\phi_o(\vec{r}, E)$ is the scalar flux $\left(\frac{\text{particles}}{cm^2 \cdot \text{Mev sec}}\right)$

$\Sigma^t(\vec{r}, E)$ is the total cross section (cm^{-1}).

This integration can be performed using spherical coordinates in a coordinate system centered at the detector point by using the transformation

$$\vec{r}' = \vec{r} - s\vec{\Omega}$$

$$dV = s^2\, ds\, d\Omega$$

$$d\Omega = d\mu\, d\theta$$

μ = the cosine of the polar angle

θ = the azimuthal angle

$$\phi_o(\vec{r}, E) = \iiint_{4\pi}\int_o^{\infty} S_o(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_o^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds\, d\Omega \qquad (2.2)$$

The most obvious reason for this transformation is that it removes the $1/s^2$ singularity.

The angular flux at any point in the geometry can be obtained from the inner spatial integration of equation 2.2:

$$\phi_o(\vec{r}, \vec{\Omega}, E) = \int_o^{\infty} S_o(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_o^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds \qquad (2.3)$$

The single scattered source density is given by an integration of the product of this angular flux and the total differential scattering cross

$$S_1(\vec{r}, \vec{\Omega}, E) = \iiint_{4\pi}\int_o^{\infty} \phi_o(\vec{r}, \vec{\Omega}', E') \frac{d^2\Sigma}{d\Omega dE}(\vec{r}, \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\, dE'\, d\Omega' \qquad (2.4)$$

where $\frac{d^2\Sigma}{d\Omega dE}(\vec{r}; \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)$

is the total differential scattering cross section (Reference 4, pg. 265). The single scattered source density yields the single scattered flux and the process is then repeated for higher order scattered sources and fluxes:

$$\phi_k(\vec{r}, \vec{\Omega}, E) = \int_o^{\infty} S_k(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_o^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds \qquad (2.5)$$

$$S_{k+1}(\vec{r}, \vec{\Omega}, E) = \iiint_{4\pi}\int_o^{\infty} \phi_k(\vec{r}, \vec{\Omega}', E') \frac{d^2\Sigma}{d\Omega dE}(\vec{r}; \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\, dE'\, d\Omega' \qquad (2.6)$$

where k is the number of scattering events which the particles *have experienced.*

The FASTER program uses random sampling techniques to perform the integrations of the order-of-scatter equations, 2.5 and 2.6, thereby giving the uncollided and scattered components of the flux. The program uses a groupwise representation of the energy dependence in conjunction with other numerical techniques--*described in Sections 3.0 through 8.0--in performing the integrations.*

The remainder of this section is a general discussion of the application of random sampling to the order-of-scatter equations. In particular, consideration will be given to the *development of sampling techniques which should minimize the error in the integration.*

In devising optimal solutions of these order-of-scatter equations, it is noted that after computing the (k-l)th and lower order flux components, the unsolved portion of the problem, corresponding to the kth and higher order flux components, *is given by a summation* over the "future" components of the flux. This unsolved portion of the problem is denoted by $\phi_{\geq k}(\vec{r}, \vec{\Omega}, E)$--the angular flux from particles having k or more collisions--and is given by a summation of equation 2.5:

$$\phi_{\geq k}(\vec{r}, \vec{\Omega}, E) = \sum_{k'=k}^{\infty} \phi_{k'}(\vec{r}, \vec{\Omega}, E)$$

$$= \int_o^{\infty} S_{\geq k}(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_o^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds \qquad (2.7a)$$

where $S_{\geq k}(\vec{r}, \vec{\Omega}, E)$ is the differential scattered source density of particles having k or more collisions.

$$S_{\geq k}(\vec{r}, \vec{\Omega}, E) = \sum_{k'=k}^{\infty} S_{k'}(\vec{r}, \vec{\Omega}, E)$$

For example, in initiating the order-of-scatter solution, the unsolved portion of the problem is:

$$\phi_{\geq o}(\vec{r},\vec{\Omega},E)=\int_o^s S_{\geq o}(\vec{r}-s\vec{\Omega},\vec{\Omega},E)\exp\left[-\int_o^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right]ds \qquad (2.7b)$$

where $\phi_{\geq o}(\vec{r},\vec{\Omega},E)$ is the total angular flux, $\phi(\vec{r},\vec{\Omega},E)$, from all orders-of-scatter and $S_{\geq o}(\vec{r},\vec{\Omega},E)$ is the total differential source density, $S(\vec{r},\vec{\Omega},E)$, including the scattered source.

In computing the kth order-of-scatter flux component, all higher order components should also be considered. If attention is given to a minimum error calculation of $\phi_k(\vec{r},\vec{\Omega},E)$, *this will not be the calculation* which minimizes the error in $\phi_{\geq k}(\vec{r},\vec{\Omega},E)$. However, it may be adequate since the two calculations are quite similar, i.e., it is theoretically possible to define kernels such that:

$$\phi_{\geq k}(\vec{r},\vec{\Omega},E)=\int_o^\infty\left[\iiint_{4\pi}\int_o^\infty S_k(\vec{r}-s\vec{\Omega},\vec{\Omega}';E')\,K(\vec{r}-s\vec{\Omega},\vec{\Omega}',E'\rightarrow\vec{r},\vec{\Omega},E)\,dE'\,d\Omega'\right]\times\exp\left[-\int_o^s \Sigma^t(\vec{r}-s\vec{\Omega},E)\,ds'\right]ds \qquad (2.8)$$

where the kernel $K(\vec{r}-s\vec{\Omega},\vec{\Omega}',E'\rightarrow\vec{r},\vec{\Omega},E)$ usually varies more slowly than the material attenuation kernel.

This "resolvent" kernel will yield the solution with an integration over the fixed source. However, this kernel is as difficult to obtain as the order-of-scatter solution is and it involves a similar iterative process, *Reference 5, pg. 522. The most well known examples* of this kernel are the dose kernels derived from moments method calculations for point isotropic sources in infinite media[6].

Use of the "resolvent" kernel is not practical for geometrically complex problems. However, the "success" of the approximate dose kernels in complicated geometries leads naturally to their use in estimating the importance of future scattering events. The use of approximate kernels is discussed in more detail in Section 2.5.

2.3 RANDOM SAMPLING CONCEPTS

The concepts involved in applying random sampling to the integration of equations 2.5 and 2.6 are simplified notationally by considering the evaluation of a simple definite integral with a non-negative integrand:

$$I = \int_{R(x)} f(x)\,dx = \int_{R(x)} \frac{f(x)}{p^*(x)}\,p^*(x)\,dx = \int_{R(x)} f^*(x)\,p^*(x)\,dx \qquad (2.9)$$

where $f^*(x) = f(x)/p^*(x)$

$R(x)$ is the range of x

$$\left.\begin{array}{l} p^*(x) \geq 0 \\ p^*(x) > 0 \text{ if } f(x) > 0 \\ \int_{R(x)} p^*(x)\,dx = 1 \end{array}\right\} \qquad (2.10)$$

The conditions imposed on $p^*(x)$ permit its use as a sampling function for obtaining e values x_i of x. This function is properly called a probability density function for the random variable x.

The mean or expected value of $f^*(x)$ is simply:

$$E\left[f^*(x)\right] = \int_{R(x)}\left[f^*(x)\right]\,p^*(x)\,dx = I \qquad (2.11)$$

The expected value of the mean square variation of $f^*(x)$ from its expected value (the variance of $f^*(x)$) is defined by:

$$\sigma^2\left[f^*(x)\right] = E\left[(f^*(x) - I)^2\right] = \int_{R(x)} \left[(f^*(x) - I)^2\right] p^*(x)\, dx = \int_{R(x)} \left[f^*(x)\right]^2 p^*(x)\, dx - I^2 = \int_{R(x)} \frac{f^2(x)}{p^*(x)}\, dx - I^2 \quad (2.12)$$

Both the mean and variance of $f^*(x)$ involve analytic integrations. Of greater interest in numerical integrations are the sample mean:

$$I_n = \frac{1}{n}\sum_{i=1}^{n} f^*(x_i),\ x_i\text{'s randomly selected from } p^*(x) \quad (2.13)$$

and the corresponding sample variance:

$$V_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left[f^*(x_i) - I_n\right]^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n}\left[f^*(x_i)\right]^2 - n I_n^2\right] \quad (2.14)$$

Simple manipulations (Reference 7, pg. 198) yield:

$$E\left[I_n\right] = I$$

$$\sigma^2\left[I_n\right] = \frac{1}{n}\sigma^2\left[f^*(x)\right]$$

$$E\left[V_n^2\right] = \sigma^2\left[f^*(x)\right].$$

The last two equations imply that:

$$\sigma^2\left[I_n\right] \approx \frac{1}{n} V_n^2 \quad (2.15)$$

Finally, it is noted that since $f(x) \geq 0$ for all x in $R(x)$, then the optimum (zero variance) integration is performed by sampling from:

$$p^*(x) = \frac{f(x)}{I} \quad (2.16)$$

since any discrete point will give the correct answer. Subsequent sections deal with approximate relationships which utilize this obvious result.

It should be noted that a transformation to the integration variable u, where $du = I^{-1} f(x)\, dx$, also yields a zero error result in a conventional numerical integration. In fact, the Monte Carlo and conventional integrations would be the same except for the technique used to obtain discrete points x_i (random versus systematic).

2.4 INNER ITERATIONS

It was possible in developing the order-of-scatter equations to explicitly write an equation for the kth order scalar flux component which involved a k-fold volume integration (and a k-fold energy integration), i.e., spatial integrations over the fixed source volume, the single scattered source volume, . . . , the kth scattered source volume. The manner in which the order-of-scatter equations was developed obviated the need for explicitly displaying these volume integrations. However, in relating the discussion of Section 2.3 to the techniques used in solving the order-of-scatter equations, this k-fold volume integration must be recognized. In particular, the discrete random variable x_i used in the preceding section is equivalent to a series of discrete position vectors $\vec{r}_{i,0}, \vec{r}_{i,1}, \vec{r}_{i,2}, \ldots, \vec{r}_{i,k}, \ldots$, obtained by random sampling.

Fortunately, the techniques used in the Monte Carlo integration of the order-of-scatter equations, i.e., the techniques used in selecting the discrete position vectors $\vec{r}_{i,k}, k = 0, 1, \ldots$, can be discussed in an orderly fashion without displaying the k-fold volume integrations. The discussion of the integrations is given below with the "outer" iteration index i of Section 2.3 suppressed. The "inner" iteration index k, corresponding to the kth order-of-scatter, will be retained.

The "order-of-scattering" inner iterants are obtained in a straight forward manner. A natural starting point is the equation for the kth component of the scalar flux:

$$\phi_k^{\cdot}(\vec{r}, E) = \iint_{4\pi} \phi_k(\vec{r}, \vec{\Omega}, E)\, d\Omega$$

$$= \iiint_{4\pi} \int_0^{\infty} S_k(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds\, d\Omega$$

$$\phi_k(\vec{r}, E) = \iiint S_k(\vec{r}', \vec{\Omega}, E) \frac{\exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right]}{s^2} dV \qquad (2.17)$$

where dV is the general differential volume element, equivalent to $s^2 ds d\Omega$ in a spherical coordinate system centered at $\vec{r}$. This equation has been transformed into a general volume integration to display the singularity $(1/s^2)$ associated with the flux calculation.

This equation has a definite value and is manipulated in the same manner as equation 2.9, i.e., the integrand is multiplied and divided by an arbitrary sampling function:

$$\phi_k(\vec{r}, E) = \iiint \left\{ \frac{S_k(\vec{r}', \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right]}{s^2\, p^*_k(\vec{r}')} \right\} p^*_k(\vec{r}')\, dV \qquad (2.18)$$

with the restrictions:

$$\left.\begin{array}{l} p^*_k(\vec{r}') \geq 0 \\ p^*_k(\vec{r}') > 0 \quad \text{if} \iiint_{4\pi} \int_0^{\infty} S_k(\vec{r}', \vec{\Omega}, E)\, dE\, d\Omega > 0 \\ \iiint p^*_k(\vec{r}')\, dV = 1 \end{array}\right\} \qquad (2.19)$$

This equation can be evaluated in a manner analogous to that used in equation 2.13 for the sample mean, i.e.: select $\vec{r}_k$ at random from $p^*_k(\vec{r})$

Then the contribution to the <u>kth</u> component of the scalar flux is given by:

$$\Delta\phi_k^*(\vec{r}, E) = \frac{S_k(\vec{r}_k, \vec{\Omega}, E)}{p^*_k(\vec{r}_k)} \; \frac{\exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right]}{s^2} \qquad (2.20)$$

where $s = |\vec{r} - \vec{r}_k|$ and $\vec{\Omega} = (\vec{r} - \vec{r}_k)/s$.

Note that the energy varies over its entire range, i.e., random sampling has been limited to the spatial variables. This is equivalent to defining an energy dependent angular point source which represents the <u>kth</u> scattered differential source density.

$$W_k^s(\vec{\Omega}, E) = \frac{S_k(\vec{r}_k, \vec{\Omega}, E)}{p^*_k(\vec{r}_k)} \qquad (2.21)$$

since the remaining factor in equation 2.20 is associated with the flux calculations and depends on the variable position vector $\vec{r}$. This point source is defined through the <u>kth</u> source distribution and can be evaluated for any desired direction $\vec{\Omega}$.

This procedure is also equivalent to representing the <u>kth</u> component of the differential source density by:

$$S_k^*(\vec{r}, \vec{\Omega}, E) = W_k^s(\vec{\Omega}, E)\, \delta(\vec{r} - \vec{r}_k) \qquad (2.22)$$

where $\delta(\vec{r} - \vec{r}_k)$ is the Dirac delta function This representation is particularly useful in later discussions where formal integrations over the <u>kth</u> scattered source volume are required.

In particular, the point representation of the fixed source (k = 0) is:

$$W_o^s(\vec{\Omega}, E) = \frac{S_o(\vec{r}_o, \vec{\Omega}, E)}{p^*_o(\vec{r}_o)} \qquad (2.23)$$

The expected value of $S_o^*(\vec{r}, \vec{\Omega}, E)$ is just the fixed source distribution:

$$E\left[W_o^s(\vec{\Omega}, E)\ \delta(\vec{r} - \vec{r}_o)\right] = \iiint_{V_o} \left[W_o^s(\vec{\Omega}, E)\ \delta(\vec{r} - \vec{r}_o)\right] P_o^*(\vec{r}_o)\, d\vec{r}_o$$

$$= \iiint_{V_o} \frac{S_o(\vec{r}_o, \vec{\Omega}, E)\ \delta(\vec{r} - \vec{r}_o)\, P_o^*(\vec{r}_o)\, d\vec{r}_o}{P_o^*(\vec{r}_o)} \qquad (2.24)$$

$$= S_o(\vec{r}, \vec{\Omega}, E)$$

The definition of the energy dependent, angular, point sources for higher order scattered components is more involved. Assuming the inner iterations have progressed through the (k-1)th spatial integration, then the (k-1)th component of the source density is represented by the vector $\vec{r}_{k-1}$ and the angular point source $W_{k-1}^s(\vec{\Omega}, E)$. Then the kth scattered energy angular point source at $\vec{r}_k$ is given by:

$$W_k^s(\vec{\Omega}, E) = \iint_{4\pi}\int_o^\infty \frac{\phi_{k-1}(\vec{r}_k, \vec{\Omega}', E')}{P_k^*(\vec{r}_k)} \frac{d^2\Sigma}{d\Omega\, dE}(\vec{r}_k, \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\, dE'\, d\Omega'$$

$$= \iint_{4\pi}\int_o^\infty \left[\frac{\int_o^\infty S_{k-1}^*(\vec{r}_k - s\vec{\Omega}', \vec{\Omega}', E') \exp\left[-\int_o^s \Sigma^t(\vec{r}_k - s'\vec{\Omega}', E')\, ds'\right] ds}{P_k^*(\vec{r}_k)}\right] \frac{d^2\Sigma}{d\Omega dE}(\vec{r}_k; \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\, dE'\, d\Omega'$$

$$= \iiint \left[\frac{\int_o^\infty W_{k-1}^s(\vec{\Omega}', E') \exp\left[-\int_o^s \Sigma^t(\vec{r}_k - s'\vec{\Omega}', E')\, ds'\right] \frac{d^2\Sigma}{d\Omega dE}(\vec{r}_k; \vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\, dE'}{s^2\, P_k^*(\vec{r}_k)}\right] \delta(\vec{r} - \vec{r}_{k-1})\, dV$$

$$= \int_o^\infty \left\{\frac{W_{k-1}^s(\vec{\Omega}_k, E') \exp\left[-\int_o^{s_k} \Sigma^t(\vec{r}_k - s'\vec{\Omega}_k, E')\, ds'\right]}{s_k^2\, P_k^*(\vec{r}_k)}\right\} \cdot \frac{d^2\Sigma}{d\Omega dE}(\vec{r}_k; \vec{\Omega}_k, E' \rightarrow \vec{\Omega}, E)\, dE' \qquad (2.25)$$

where $s_k = |\vec{r}_k - \vec{r}_{k-1}|$ and $\vec{\Omega}_k = (\vec{r}_k - \vec{r}_{k-1})/s_k$

Note that the (k-1)th source density component was replaced by its equivalent point representation $W_{k-1}^s(\vec{\Omega}, E)$. Thus, this equation is an equality in the sense that the expected value of the right hand side is the left hand side.

It now becomes expedient to define an energy dependent point monodirectional quantity at $\vec{r}_k$ which characterizes the (k-1)th component of the flux:

$$W_{k-1}^\phi(E) = W_{k-1}^s(\vec{\Omega}_k, E) \frac{\exp\left[-\int_o^{s_k} \Sigma^t(\vec{r}_k - s'\vec{\Omega}_k, E)\, ds'\right]}{s_k^2\, P_k^*(\vec{r}_k)} \qquad (2.26)$$

The factor associated with random selection of the discrete position vector $\vec{r}_k$ has been included in this energy-dependent, point monodirectional representation of the flux to simplify the final equation for the kth scattered, angular point source:

$$W_k^s(\vec{\Omega}, E) = \int_o^\infty W_{k-1}^\phi(E') \frac{d^2\Sigma}{d\Omega dE}(\vec{r}_k; \vec{\Omega}_k, E' \rightarrow \vec{\Omega}, E)\, dE' \qquad (2.27)$$

The order of calculations may clarify this procedure:

a) select $\vec{r}_k$ from $P_k^*(\vec{r}')$

b) calculate $s_k = |\vec{r}_k - \vec{r}_{k-1}|$

c) calculate $\vec{\Omega}_k = (\vec{r}_k - \vec{r}_{k-1})/s_k$

d) calculate $W_{k-1}^s(\vec{\Omega}_k, E)$ for the discrete direction $\vec{\Omega}_k$

$k = 1$, $W_o^s(\vec{\Omega}_k, E)$: from equation 2.23 for $\vec{\Omega} = \vec{\Omega}_k$

$k > 1$, $W_{k-1}^s(\vec{\Omega}_k, E)$ from equation 2.27 for $\vec{\Omega} = \vec{\Omega}_k$ using $W_{k-2}^\phi(E)$.

e) calculate $W_{k-1}^\phi(E)$ using equation 2.26.

f) then $W_k^s(\vec{\Omega}, E)$ can be calculated, as required for a given direction $\vec{\Omega}$, by using equation 2.27.

Thus, $W_k^s(\vec{\Omega}, E)$ is not computed directly except as required for discrete directions. It is always available through the fixed source distribution for k = 1, or, for k > 1, through $W_{k-1}^{\phi}(E)$ and the differential scattering cross sections. This is indicated in step d) above.

For k = 1, the verification of the representation of the source given by equation 2.27 is obtained by calculating expected values:

$$E\left[W_1^s(\vec{\Omega},E)\,\delta(\vec{r}-\vec{r}_1)\right] = \iiint_{V_0} \iiint_{V_1} \left[W_1^s(\vec{\Omega},E)\,\delta(r-r_1)\right] p_1^*(\vec{r}_1)\, p_0^*(\vec{r}_0)\, d\vec{r}_1\, d\vec{r}_0$$

$$= \iiint_{V_0} \iiint_{V_1} \left\{ \int_0^\infty \frac{S_0(\vec{r}_0,\vec{\Omega}_1,E')\exp\left[-\int_0^{s_1}\Sigma^t(\vec{r}_1 - s'\vec{\Omega}_1, E')\,ds'\right]}{s_1^2\, p_0^*(\vec{r}_0)\, p_1^*(\vec{r}_1)} \right.$$

$$\left. \times \frac{d^2\Sigma}{d\Omega dE}(\vec{r}_1; \vec{\Omega}_1, E' \rightarrow \vec{\Omega}, E)\, dE' \right\} \delta(\vec{r}-\vec{r}_1)\, p_1^*(\vec{r}_1)\, p_0^*(\vec{r}_0)\, d\vec{r}_1\, d\vec{r}_0$$

$$= \iiint_{V_0} \left\{ \int_0^\infty S_0(\vec{r}_0, \vec{\Omega}', E') \frac{\exp\left[-\int_0^s \Sigma^t(\vec{r} - s\vec{\Omega}', E')\,ds'\right]}{s^2} \frac{d^2\Sigma}{d\Omega dE}(\vec{r};\vec{\Omega}', E' \rightarrow \vec{\Omega}, E)\, dE' \right\} d\vec{r}_0$$

$$= S_1(\vec{r}, \vec{\Omega}, E) \tag{2.28}$$

Verification for higher order components can be obtained by induction, i.e., by assuming:

$$E\left[W_k^s(\vec{\Omega}, E)\,\delta(\vec{r}-\vec{r}_k)\right] = \iiint_{V_0} \cdots \iiint_{V_k} W_k^s(\vec{\Omega}, E)\,\delta(\vec{r}-\vec{r}_k)\, p_k^*(\vec{r}_k) \cdots p_0^*(\vec{r}_0)\, d\vec{r}_k \cdots d\vec{r}_0$$

$$= S_k(\vec{r}, \vec{\Omega}, E)$$

and then showing that

$$E\left[W_{k+1}^s(\vec{\Omega}, E)\,\delta(\vec{r}-\vec{r}_{k+1})\right] = S_{k+1}(\vec{r}, \vec{\Omega}, E)$$

Similar verifications can also be obtained for arbitrary point kernels, e.g., those used for flux calculations (Section 2.6). The resulting integrations are similar to those above and are not shown.

The final set of procedures used in a single outer iteration is shown in Figure 2. The inner iterations cannot proceed indefinitely as is indicated by the order-of-scatter solution. Therefore, the inner iterations are terminated using various criteria such as total number of iterations (collision cutoff), all energies below some minimum (energy cutoff), or all flux contributions being negligible (weight cutoff).

2.5 SAMPLING FUNCTIONS

At this point, the definitions of the sampling or probability density functions are arbitrary except for the conditions imposed in Equation 2.19. The criterion selected for defining optimal functions is that the variance in the contributions from all future inner iterations be minimum.

After (k-1) inner iterations, the scalar flux will have been estimated to the extent of:

$$\phi_{<k}(\vec{r}, E) = \sum_{k'=0}^{k-1} \phi_{k'}(\vec{r}, E) \tag{2.29}$$

The remainder associated with neglecting higher order components of the flux or, alternatively, the importance of these components is given by:

$$\phi_{\geq k}(\vec{r}, E) = \sum_{k'=k}^{\infty} \phi_{k'}(\vec{r}, E) \tag{2.30}$$

Since a detailed treatment of these "future" components is as difficult as solving the original problem--equation 2.7a would be solved--various approximations must be made. It is noted

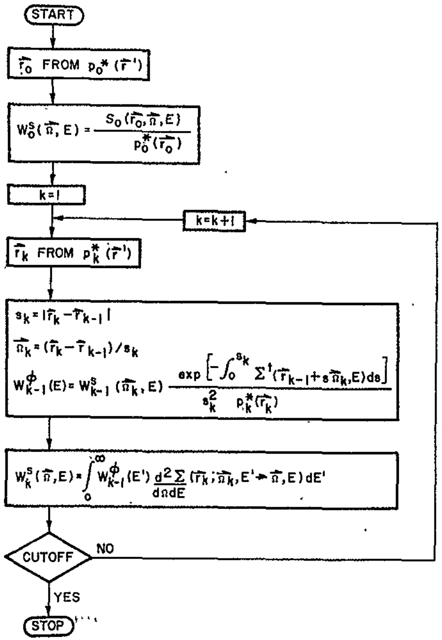


Figure 2. The Monte Carlo Method - Inner Iterations

Astronuclear Laboratory

that a single point criterion--independent of energy--is more easily applied, and much easier to estimate; e.g.

$$D_k(\vec r) = \int_0^\infty f(E)\,\phi_k(\vec r, E)\,dE \qquad (2.31)$$

where f(E) is a response function of particular importance to the problem being solved, e.g., the flux to dose conversion factor if dose rates are being calculated. The solution and remainder corresponding to this response function are then given by:

$$D_{<k}(\vec r) = \sum_{k'=0}^{k-1} D_{k'}(\vec r) \quad \text{(Total response already calculated)} \qquad (2.32)$$

$$D_{\geq k}(\vec r) = \sum_{k'=k}^{\infty} D_{k'}(\vec r) \quad \text{(uncalculated contributions to total response)} \qquad (2.33)$$

The calculation of the latter quantity is also as difficult as solving the original problem. However, for estimating purposes, the "point-kernel method" is available for approximating the remainder

$$D_{\geq k}(\vec r) = \sum_{k'=k}^{\infty} \int_0^\infty f(E)\,\phi_{k'}(\vec r, E)\,dE$$

$$= \sum_{k'=k}^{\infty} \int_0^\infty f(E) \iiint S_{k'}(\vec r\,', \vec\Omega, E)\,\frac{\exp\left[-\int_0^s \Sigma^t(\vec r - s'\,\vec\Omega, E)\,ds'\right]}{s^2}\,dV dE$$

$$= \iiint \frac{\left\{\int_0^\infty \left[\sum_{k'=k}^{\infty} S_{k'}(\vec r\,', \vec\Omega, E)\right] \exp\left[-\int_0^s \Sigma^t(\vec r - s'\,\vec\Omega, E)\,ds'\right] f(E)\,dE\right\}}{s^2}\,dV$$

$$\approx \tilde D_{\geq k}(\vec r) \qquad (2.34)$$

where

$$\tilde{D}_{\geq k}(\vec{r}) = \iiint \left\{ \int_0^{\infty} \iint_{4\pi} S_k(\vec{r}', \vec{\Omega}', E)\, K(\vec{\Omega}', E, \eta(E)) \exp\left[-\eta(E)\right] f(E)\, d\Omega' dE \right\} \frac{dV}{s^2} \quad (2.35)$$

$$s = |\vec{r} - \vec{r}'|, \vec{\Omega} = (\vec{r} - \vec{r}')/s$$

$$\eta(E) = \int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E) ds'$$

$k(\vec{\Omega}', E, \eta(E))$ is some approximate representation of the response contributions by multiple scattering events, usually a simple function of the number of the mean-free-paths $\eta(E)$ between a source (or scattering) point and the detector point.

Equation 2.35 not only estimates the importance of future scattering events but can also be used to estimate the importance of the "source", or scattering points. Since it is a feasible spatial importance estimator -- it is related in an approximate manner to the zero variance importance function -- and since it lends itself to a variety of further approximations, it plays an important role in the development of optimal sampling schemes.

Equation 2.35 can be manipulated in a manner similar to equation 2.18:

$$\tilde{D}_{\geq k}(\vec{r}) = \iiint \frac{\left[\int_0^{\infty} \iint_{4\pi} S_k(\vec{r}', \vec{\Omega}', E)\, K(\vec{\Omega}', E, \eta(E)) \exp\left[-\eta(E)\right] f(E) d\Omega' dE\right]}{s^2 p_k^*(\vec{r}')} p_k^*(\vec{r}') dV$$

Equation 2.16 then implies that the optimum sampling function, yielding zero variance for $\tilde{D}_{\geq k}(\vec{r})$, is given by:

$$p_k^*(\vec{r}') \quad \frac{1}{\tilde{D}_{\geq k}(\vec{r})} \frac{\int_0^{\infty} \iint_{4\pi} S_k(\vec{r}', \vec{\Omega}', E)\, K(\vec{\Omega}', E, \eta(E)) \exp\left[-\eta(E)\right] f(E) d\Omega' dE}{s^2} \quad (2.36a)$$

This equation is still quite involved and its numerical implementation requires further simplification and approximation.

The form of this equation may be clarified by examining its equivalent for a one velocity problem. Assuming a uniform infinite medium and isotropic sources, either fixed or scattered, this equation can be approximated by:

$$p_k^*(\vec{r}') = \frac{S_k(\vec{r}') \exp\left[-\Sigma^r s\right]/s^2}{\iiint S_k(\vec{r}') \frac{\exp\left[-\Sigma^r s\right]}{s^2} dV} \quad (2.36b)$$

where Σ^r is an effective cross section giving the attenuation characteristics of the total flux from the kth scattered source and $S_k(\vec{r}')$ is the differential kth scattered source density.

For point detectors, random sampling of the fixed source distribution, $k = 0$, can utilize equation 2.36 directly. The more important aspects of this equation are:

a) it includes the spatial divergence ($1/s^2$) from the detector;

b) it retains the exponential falloff of source point importance due to material attenuation, and

c) it includes the fixed source distribution.

The techniques used in the FASTER program for sampling the fixed source are closely associated with equation 2.36b. An average source energy is used to define the necessary sampling parameters. The program includes, however, an explicit representation of the material distribution. The details of the sampling procedures are discussed in Section 8.3.

Applications of equation 2.36 to higher order flux components, $k > 0$, involves the scattered source definition using the (k-1)th point source representation:

$$p_k^*(\vec{r}') = \frac{1}{\tilde{D}_{\geq k}(\vec{r})} \int_0^{\infty} \iint_{4\pi} \int_0^{\infty} W_{k-1}^s(\vec{\Omega}', E') \frac{d^2\Sigma}{d\Omega\, dE}(\vec{r}'; \vec{\Omega}', E' \to \vec{\Omega}, E)$$

$$\frac{K(\vec{\Omega}', E, \eta(E)) \exp\left[-\int_0^t \Sigma^t(\vec{r}' - t'\vec{\Omega}', E) dt' - \eta(E)\right] f(E) dE' d\Omega' dE}{t^2 s^2} \quad (2.37a)$$

where $t = |\vec{r}' - \vec{r}_{k-1}|$, $\vec{\Omega}' = (\vec{r}' - \vec{r}_{k-1})/t$

Again, a large degree of approximation is required for numerical calculations.

A one-velocity approximation may also clarify this equation. Using the assumptions used in equation 2.36, the sampling function can be approximated by:

$$P_k^* (\vec{r}') = \frac{\dfrac{\exp\left[-\Sigma^t t - \Sigma^r s\right]}{t^2 s^2}}{\displaystyle\iiint \frac{\exp\left[-\Sigma^t t - \Sigma^r s\right]}{t^2 s^2} dV} \qquad (2.37b)$$

where Σ^t is the total cross section.

For a point detector the following characteristics are noted for this scattering point sampling function:

a) it includes the spatial divergence from both the source $(1/t^2)$ and the detector $(1/s^2)$

b) it includes the exponential attenuation along both "legs" of the scattering triangle

c) it includes the spatial and angular dependence of the scattering cross section, and

d) it includes the angular dependence of the (k-1)th source component.

The FASTER program uses a one-velocity approximation of this sampling function with group averaged parameters being obtained at each scattering point. The material distribution and scattering angle effects are included by several alternate approximations. The details are discussed in Sections 8.4, 8.5, and 8.6.

Similar equations can be obtained for sampling functions which minimize the variance of volume and surface averaged flux calculations. However, these equations involve an integration over the spatial extent of these volumes and surfaces and the development is more complicated. These sampling functions have been approximated, therefore, by "solid angle" considerations as described in Section 8.0.

2.6 POINT ANGULAR FLUXES

The set of inner iterations yields a single estimate of the total differential source density:

$$S_i^* (\vec{r}, \vec{\Omega}, E) = \sum_{k=0} W_{ik}^s (\vec{\Omega}, E) \; \delta(\vec{r} - \vec{r}_{ik}) \qquad (2.38)$$

where the index i corresponds to the ith repetition of this process, or a single outer iteration. Previous discussions verified that:

$$E\left[S_i^* (\vec{r}, \vec{\Omega}, E)\right] = S(\vec{r}, \vec{\Omega}, E) - R_i \qquad (2.39)$$

where R_i is a remainder corresponding to the termination of the inner iterations after a finite value of k, i.e., the neglected higher order-of-scatter components.

Repetitive application of these techniques (n outer iterations) yields the final estimate of the total differential source density:

$$S^* (\vec{r}, \vec{\Omega}, E) = \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{k=0} W_{ik}^s (\vec{\Omega}, E) \; \delta(\vec{r} - \vec{r}_{ik})\right] \qquad (2.40)$$

This is not the end result, of course. The original intent was to obtain the flux at an arbitrary point. Equation 2.7b implies the total angular flux is obtained by the integration.

$$\phi(\vec{r}, \vec{\Omega}, E) = \int_0^\infty S(r - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t (\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds \qquad (2.41)$$

This can be approximated using the above source density:

$$\phi^* (\vec{r}, \vec{\Omega}, E) = \int_0^\infty S^*(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E)\, ds'\right] ds \qquad (2.42)$$

It is a straightforward task to show that the expected value of this approximation is, indeed, the total angular flux within an error $\mathcal{R}$ corresponding to the source errors R_i above:

$$E\left[\phi^*(\vec{r},\vec{\Omega},E)\right] = \phi(\vec{r},\vec{\Omega},E) - \mathcal{R} \tag{2.43}$$

The verification involves integrations similar to those used to verify the representation of the differential source density, equations 2.24 and 2.28, and is not shown.

Substitution of equation 2.40 into equation 2.42 yields:

$$\begin{aligned}\phi^*(\vec{r},\vec{\Omega},E) &= \int_0^\infty \left[\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{k=0} W^s_{ik}(\vec{\Omega},E)\,\delta(\vec{r}-\vec{r}_{ik})\right]\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right]\right] ds \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}\int_0^\infty W^s_{ik}(\vec{\Omega},E)\,\delta(\vec{r}-\vec{r}_{ik})\,\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right] ds\end{aligned} \tag{2.44}$$

where the order of the summations and integration has been reversed. To eliminate some of the notational clutter, the individual elements in the summation will be examined with the outer iteration index suppressed:

$$\Delta\phi^*_k(\vec{r},\vec{\Omega},E) = \int_0^\infty W^s_k(\vec{\Omega},E)\,\delta(\vec{r}-\vec{r}_k)\,\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right]\cdot ds \tag{2.45}$$

This is also the procedure used in numerical calculations, i.e., individual contributions are computed without considering the iteration index.

The spatial integration is performed using the relationship:

$$\delta(\vec{r}-s\Omega-\vec{r}_k) = \delta(s-s_k)\,\frac{\delta(\vec{\Omega}-\vec{\Omega}_k)}{s_k^2} \tag{2.46}$$

where:

$$s_k = \left|\vec{r}-\vec{r}_k\right|, \quad \vec{\Omega}_k = (\vec{r}-\vec{r}_k)/s_k$$

Thus:

$$\begin{aligned}\Delta\phi_k^*(\vec{r},\vec{\Omega},E) &= \int_0^\infty W_k^s(\vec{\Omega},E)\,\delta(s-s_k)\,\frac{\delta(\vec{\Omega}-\vec{\Omega}_k)}{s_k^2}\,\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right] ds \\ &= W_k^s(\vec{\Omega},E)\,\delta(\vec{\Omega}-\vec{\Omega}_k)\,\frac{\exp\left[-\int_0^{s_k}\Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right]}{s_k^2}\end{aligned}$$

The delta function involving the direction vector complicates nothing. In particular, the corresponding contribution to the scalar flux is given by an integration over solid angle:

$$\begin{aligned}\Delta\phi_k^*(\vec{r},E) &= \iint_{4\pi}\Delta\phi_k^*(\vec{r},\vec{\Omega},E)\,d\Omega \\ &= W_k^s(\vec{\Omega}_k,E)\cdot\frac{\exp\left[-\int_0^{s_k}\Sigma^t(\vec{r}-s'\vec{\Omega}_k,E)\,ds'\right]}{s_k^2}\end{aligned} \tag{2.48}$$

as one would expect.

As indicated in Section 2.4, the sampling functions which minimize the variance in the flux at the point $\vec{r}$, involve this point in a rather complicated manner. However, the techniques used in flux estimation are independent of these considerations and can be used for an arbitrary set of points. It should be noted that flux estimation for arbitrary points will yield a result with an infinite variance[8] unless these points are located in volumes which exclude source and scattering points. That is, the $1/s_k^2$ factor is the trouble maker and its deleterious effect can only be removed by including it in the sampling function or by excluding small values of s_k.

2.7 SPATIALLY AVERAGED ANGULAR FLUXES

The problem of infinite variance flux estimates can also be removed by averaging the fluxes at arbitrary points over a specified surface or volume. While these averaged results are usually less desirable than a set of point results, there are instances when averages are the only requirement.

The contribution to a surface averaged flux is obtained by integrating the point result given by equation 2.47:

$$\Delta\phi_k^*(\vec{\Omega}, E)_A = \frac{1}{A}\iint_A W_k^s(\vec{\Omega}, E)\,\delta(\vec{\Omega}-\vec{\Omega}_k)\,\frac{\exp\left[-\int_0^{s_k}\Sigma^t(\vec{r}-s'\vec{\Omega}, E)\,ds'\right]dA}{s_k^2} \qquad (2.49)$$

where A is the area of the specified surface, dA is a differential element of area, and $\vec{r}$ is a point on the surface. The integration is than transformed to an integration over solid angle about the point $\vec{r}_k$:

$$\Delta\phi_k^*(\vec{\Omega}, E)_A = \frac{1}{A}\iint_{4\pi} W_k^s(\vec{\Omega}, E)\,\delta(\vec{\Omega}-\vec{\Omega}_k)\,\frac{\exp\left[-\int_0^{s_k}\Sigma^t(\vec{r}-s'\vec{\Omega}, E)\,ds'\right]}{s_k^2}\,\frac{s_k^2\,d\Omega_k}{|\vec{\Omega}_k\cdot\vec{n}|} \qquad (2.50)$$

where $dA = s_k^2\,d\Omega_k/|\vec{\Omega}_k\cdot\vec{n}|$, $\vec{n}$ is the unit normal to the surface at $\vec{r}$, and a summation over multiple points on the surface which yield the same direction vector $\vec{\Omega}_k$ is implicit. Note that the integration is performed using a differential element of solid angle $d\Omega_k$ since $\vec{\Omega}_k$ is the direction vector determining the point (s) $\vec{r}$ on the surface. Using the reciprocal nature of the Dirac delta function and the material attenuation kernel, the integration yields:

$$\Delta\phi_k^*(\vec{\Omega}, E)_A = \frac{1}{A}W_k^s(\vec{\Omega}, E)\,\frac{\exp\left[-\int_0^{s(\vec{\Omega})}\Sigma^t(\vec{r}_k+s'\vec{\Omega}, E)\,ds'\right]}{|\vec{\Omega}\cdot\vec{n}|} \qquad (2.51)$$

where $s(\vec{\Omega})$ is the distance to the surface.

The contribution to a volume averaged flux is also obtained by integrating the point result:

$$\Delta\phi_k^*(\vec{\Omega}, E)_V = \frac{1}{V}\iiint_V W_k^s(\vec{\Omega}, E)\,\delta(\vec{\Omega}-\vec{\Omega}_k)\,\frac{\exp\left[-\int_0^{s_k}\Sigma^t(\vec{r}-s\vec{\Omega}, E)\,ds'\right]dV}{s_k^2} \qquad (2.52)$$

where V is the volume over which the flux is averaged.

The integration is transformed using a spherical coordinate system centered at $\vec{r}_k$:

$$\Delta\phi_k^*(\vec{\Omega}, E)_V = \frac{1}{V}\iint_{4\pi}\int_{s\text{ in }V} W_k^s(\vec{\Omega}, E)\,\delta(\vec{\Omega}-\vec{\Omega}_k)\,\frac{\exp\left[-\int_0^{s_k}\Sigma^t(r_k+s\vec{\Omega}, E)\,ds'\right]}{s_k^2}\,s_k^2\,ds_k\,d\Omega_k \qquad (2.53)$$

where $dV = s_k^2\,ds_k\,d\Omega_k$ since $\vec{\Omega}_k$ and s_k define the points $\vec{r}$ in the volume.

Using the same arguments as before, the integration yields:

$$\Delta\phi_k^*(\vec{\Omega}, E) = \frac{1}{V} W_k^s(\vec{\Omega}, E) \int_{s \text{ in } V} \exp\left[-\int_0^s \Sigma^t(\vec{r}_k + s'\vec{\Omega}, E)\, ds'\right] ds \tag{2.54}$$

The integration over distance s, is limited to points $\vec{r}$ on $\vec{r}_k + s\vec{\Omega}$ which lie in the volume. There may, of course, be several discrete intersections with the volume along $\vec{\Omega}$.

There are two interesting, special forms of this equation. Considering only one intersection with the volume, let $s(\vec{\Omega})$ be the distance to the volume and $\Delta s(\vec{\Omega})$ the distance in the volume. Then, if the volume is void:

$$\Delta\phi_k^*(\vec{\Omega}, E) = \frac{1}{V} W_k^s(\vec{\Omega}, E) \exp\left[-\int_0^{s(\vec{\Omega})} \Sigma^t(\vec{r}_k + s'\vec{\Omega}, E)\, ds'\right] \Delta s(\vec{\Omega}) \tag{2.55}$$

and if the volume has constant material properites:

$$\Delta\phi_k^*(\vec{\Omega}, E)_V = \frac{1}{V} W_k^s(\vec{\Omega}, E) \exp\left[-\int_0^{s(\vec{\Omega})} \Sigma^t(\vec{r}_k + s'\vec{\Omega}, E) ds'\right] \frac{\left[1 - \exp\left[-\Delta s(\vec{\Omega})\Sigma^V(E)\right]\right]}{\Sigma^V(E)} \tag{2.56}$$

where $\Sigma^V(E)$ is the total cross section at any point in the volume.

There is a possible difficulty with both the surface and volume averaged flux contributions since they are still defined for all directions $\vec{\Omega}$; i.e., a direct numerical integration to obtain various flux components could be prohibitive. If so, random sampling can be

used. A general equation for the integration of an angular kernel $g(\vec{\Omega})$ is written first:

$$\Delta G_k^*(E) = \iint_{4\pi} \Delta\phi_k^*(\vec{\Omega}, E)_{\begin{Bmatrix}A\\V\end{Bmatrix}}\, g(\vec{\Omega})\, d\Omega \tag{2.57}$$

$$= \iint_{4\pi} \left\{ \frac{\Delta\phi_k^*(\vec{\Omega}, E)}{q^*(\vec{\Omega})} \begin{Bmatrix}A\\V\end{Bmatrix} g(\vec{\Omega}) \right\} q^*(\vec{\Omega})\, d\Omega$$

$$\approx \frac{1}{L} \sum_{\ell=1}^{L} \frac{\Delta\phi_k^*(\vec{\Omega}_\ell, E)}{q^*(\vec{\Omega}_\ell)} \begin{Bmatrix}A\\V\end{Bmatrix} g(\vec{\Omega}_\ell) \tag{2.58}$$

where $\begin{Bmatrix}A\\V\end{Bmatrix}$ denotes either a surface or volume,

$$\left.\begin{aligned} & q^*(\vec{\Omega}) \geq 0 \\ & q^*(\vec{\Omega}) > 0 \text{ if } \int_0^\infty \Delta\phi_k^*(\vec{\Omega}, E) \begin{Bmatrix}A\\V\end{Bmatrix} dE > 0 \\ & \iint_{4\pi} q^*(\vec{\Omega})\, d\Omega = 1 \end{aligned}\right\} q^*(\vec{\Omega}) \text{ is a sampling function} \tag{2.59}$$

L is the total number of discrete directions, and $\vec{\Omega}_\ell$ is a discrete direction obtained by random sampling of $q^*(\vec{\Omega})$.

This is equivalent to representing the averaged angular fluxes by:

$$\Delta\phi_k^*(\vec{\Omega}, E)_A = \frac{1}{A} \frac{1}{L} \sum_{\ell=1}^{L} \frac{W_k^s(\vec{\Omega}, E)}{q^*(\vec{\Omega}_\ell)} \frac{\exp\left[-\int_0^{s(\vec{\Omega})} \Sigma^t(\vec{r}_k + s'\vec{\Omega}, E) ds'\right]}{|\vec{\Omega} \cdot \vec{n}|} \delta(\vec{\Omega} - \vec{\Omega}_\ell) \tag{2.60}$$

$$\Delta\phi_k^*(\vec{\Omega},E)_V = \frac{1}{V}\frac{1}{L}\sum_{\ell=1}^{L}\frac{W_k^s(\vec{\Omega},E)}{q^*(\vec{\Omega}_\ell)}\int_{s \text{ in } V} \exp\left[-\int_0^s \Sigma^t(\vec{r}_k+s'\vec{\Omega},E)ds'\right]ds \;\; \delta(\vec{\Omega}-\vec{\Omega}_\ell) \qquad (2.61)$$

for surfaces and volumes respectively since angular integrations with the arbitrary kernél $g(\vec{\Omega})$ yield equation 2.58 above.

The definition of $q^*(\vec{\Omega})$ which will minimize the error in the averaged flux angular integrations can be argued in a manner similar to the arguments used in defining the functions $p_k^*(\vec{r})$ used in selecting the source and scattering points. The arguments are simplified since future scattering contributions need not be considered--this is just an angular integration. The arguments are complicated by the fact that the same set of random discrete directions will probably (not necessarily) be used for all the surfaces and/or volumes over which fluxes are being averaged and these surfaces and/or volumes may occupy widely varying spatial positions.

2.8 CONCLUDING REMARKS

The previous sections pertained to a development of the Monte Carlo method which utilized random sampling for all of the spatial integrations. This is the procedure used in the FASTER program. In particular, the major equations used in the program are the numerical equivalents of:

a) equation 2.36 for selecting source points,
b) equation 2.23 for the point representation of the source,
c) equation 2.37 for selecting scattering points,
d) equation 2.26 for the point representation of the flux of particles going into a collision,
e) equation 2.27 for the point representation of scattered sources,
f) equation 2.47 for angular point flux estimation,
g) equation 2.60 for surface averaged angular flux estimation, and
h) equations 2.61, in conjunction with equations 2.55 and 2.56, for volume averaged angular flux estimation.

Each integration involved in calculating the order-of-scatter fluxes could have been performed by either random sampling or by conventional numerical integrations. In particular, most Monte Carlo programs use random sampling for the energy integrations as well as the spatial integrations and the entire calculational procedure is equivalent to the simulation of individual particle histories.[9]

A variety of other combinations of integration techniques can be used. In particular, the variance associated with selecting discrete points from the fixed source in a Monte Carlo integration can be replaced by the systematic error involved in a conventional numerical integration over the spatial extent of the source volume. In fact, uncollided flux calculations can use various combinations of random sampling and direct numerical integration for the three spatial variables. One combined integration procedure is discussed in Appendix A.

3.0 PROGRAM LOGIC

The techniques described in Section 2.0 permit a large degree of separation of conventional numerical techniques from random sampling techniques. This separability is utilized in the structure of the FASTER program through a series of subprograms that perform conventional calculations, such as, source interpolation at a point and a single scattering calculation for a fixed scattering angle. Another series of subprograms are used in the random selection of the parameters, e.g., the source point or scattering angle, for these conventional calculations.

3.1 DATA REQUIREMENTS

Several major divisions in the data required by the FASTER program have been made. Detailed input instructions for these data are given in Section 9.0.

Section 1 data, i.e., data in the first section of input, include the limits and controls for the FASTER calculations. The requisite data are described in Section 9.2.

Section 2 data are used in describing the geometry of the problem. Details of the geometric calculations performed by FASTER are given in Section 4.0. Data input instructions for geometric parameters are given in Section 9.3.

Section 3 data involve the description of the distributed sources. Calculations associated with these sources are described in Section 5.0. Input instructions for the description of sources are given in Section 9.4.

Section 4 data include the microscopic cross sections used to obtain the macroscopic attenuating and scattering properties of the non-void regions of the geometry. The calculations requiring these cross sections are described in Section 6.0. Detailed input instructions for the cross sections are given in Section 9.5.

Section 5 data are used to specify the final form of the computed results. This includes the description of the various points, surfaces and/or volumes for which the FASTER program will compute multigroup fluxes. Details of the various flux component estimates used in FASTER are given in Section 7.0. Input instructions for this section of data are given in Section 9.6.

Section 6 data pertain to the description of the random sampling functions. The sampling functions incorporated in FASTER are described in Section 8.0. Input instructions for the sampling parameters are given in Section 9.7.

3.2 PROGRAM FLOW

This section attempts to clarify the inter-relationships of data and calculational techniques by discussing the general flow within the FASTER program. The major subprograms of FASTER are also identified as to their function, i.e., which equations they contain.

The program is divided into two major parts. The first part involves data input and preparation and is controlled by the subroutine DEFINE. The second part involves the actual calculations and these are controlled by the subroutine SOLVIT.

Data Input and Preparation

DEFINE is the first routine called by FASTER. It calls other routines, in order, for the input of the data described in Section 9.0. The first subroutine entered is STORER. This routine reads the Section 1 data inputs and then allocates storage for all the dimensioned data. If multiple cases are run, it also manipulates the data arrays to account for dimension changes, etc.

Subroutine GEOMIN is then entered and all Section 2 data are input. After all geometric inputs have been processed, the ± sign associated with the boundaries of the regions are calculated using the input coordinates of an arbitrary point in each region. Then the geometry is checked using the input point-in-region coordinates and the function subprogram LOCATE, which computes the region(s) occupied by arbitrary points.

Section 3 data are input next in subroutine SOURCE. This includes the energy group structure and the definition and normalization of all the sources. Subroutine INSECT accepts Section 4 cross section data and combines the microscopic data into the requisite macroscopic data.

Section 5 data are input in subroutine RESULT. This includes the flux groups, flux conversion factors, and detector definitions.

Finally, the input sampling parameters--Section 6 data--are input by subroutine RANDOM.

After all data are input, a check of an error count is made. If any errors were detected, the data for the next case is input. If no errors were detected, control is passed back to the FASTER program.

Calculations

With all requisite data well defined, control is passed from FASTER to SOLVIT. The SOLVIT routine passes control to one of two available calculational control subprograms. The first, SOBER, was written to compute surface and volume averaged fluxes and/or fluxes at multiple point detectors in void regions. The second control subprogram, SOLVER, was coded for the individual treatment of point detectors at arbitrary locations in the geometry.

As indicated, these routines perform almost identical functions. The following discussion attempts to describe both simultaneously. There may be minor variations from the actual order of some of the calculations, but they are unimportant in the overall picture. Some of the routines discussed below also require computations by other routines, however, this secondary control level will not be discussed.

The first step is the definition of a preferred point for use in the sampling procedures. In SOBER it is defined by input and is surrounded by a sphere with an input radius which encloses a volume in space where fluxes will be calculated. SOLVER contains an iteration over detector points and the preferred point is the detector point being treated. An average source group index, used in a one-velocity approximation of equation 2.36, is then computed by subroutine GROUP.

All other calculations are performed within the outer iteration loop. The first calculation within this loop is the random selection of a position vector $\vec{r}_0$ from th- fixed source. This is done by one of two subroutines; PSTAR, if the sampling is performed in the source geometry coordinate system or SPHERE, which approximates equation 2.36.

The calculations enter the inner iteration loop where they remain until terminated by one of the cutoff criteria, i.e., maximum number of inner iterations (collision cutoff), weight cutoff, or energy cutoff. The first iteration in this inner loop differs from all subsequent inner iterations because the source point is actually a point in a source volume. For all inner iterations after the first, it is actually a scattering (scattered source) point.

The first calculation in the inner loop involves the flux contribution to the point detectors. For each of these detectors, calculation of the distance and direction to the detector from the source point is performed by the function subprogram VECTOR. The calculation of the angular point source for the direction towards the detector is then performed by interpolation in the SZERO subroutine (equation 2.23) if this is the first inner iteration, or by SINGLE (equation 2.27) for subsequent inner iterations. If there is a non-zero source for this direction, the path lengths through the various regions lying between the source (scattering) point and the detector point are computed by subroutine PATH. The mean-free-paths *(mfp) along the total path are then computed for each source group by subroutine* KERNEL and the flux estimation performed in subroutine DETECT using equation 2.47.

The next step is performed only in the SOBER inner iterations. It involves the random selection of discrete directions for calculating surface and volume averaged fluxes. These directions are obtained from subroutine VSTAR. A possible exception is the first inner iteration where they can be obtained from subroutine QSTAR. The source for each fixed direction is computed using SZERO or SINGLE. If non-zero, the regions lying along the direction vector are computed by PATH. Each of the regions lying along the direction vector is checked to see if it is a volume detector and should receive a flux contribution. If so, the mfp's up to the region are computed by KERNEL and the flux is computed by DETECT using contribution equation 2.61 in conjunction with equations 2.55 and 2.56. Each boundary crossing between regions is also checked to see if it is a surface detector. If it is, the normal derivative at the

boundary is computed by subroutine NORMAL; the mfp's to the boundary are obtained from KERNEL and the flux contributions are then computed by DETECT using equation 2.60.

The final step in each inner iteration is the random selection of the next scattering point. The average flux contribution energy, as calculated by DETECT, is used to define an average energy group for the sampling procedures. The direction vector defining the scattering point is then obtained from VSTAR (or possibly QSTAR on the first inner iteration). The regions lying along the ray defined by this direction are computed by PATH. The distance to the collision point is obtained from USTAR (equation 2.37) and the previous source point is then evaluated for the direction vector from this prior point to the new point, using SZERO or SINGLE. If non-zero, the mfp's to the scattering point are obtained from KERNEL, and the point monodirectional fluxes are computed at the scattering point using equation 2.26. *The next inner iteration is initiated with the sources being obtained from SINGLE using the* monodirectional fluxes. These inner iterations are continued until a cutoff is obtained.

The outer iterations are continued to a specified maximum with a printout of the flux edits being performed by subroutine ANSWER at specified intervals.

SECTION
4.0 GEOMETRIC CONSIDERATIONS

The most important feature of the Monte Carlo method--in comparison with other "exact" solutions of the transport equation--is its applicability to complicated geometries. This feature is implemented in the FASTER program by utilizing the general quadric surface equation. The numerical analysis presented below follows that of Reference 10.

4.1 QUADRIC SURFACES

The general quadric equation for a specified surface i is:

$$\begin{aligned} u_i(\vec{r}) = a_{o,i} &+ a_{1,i}x + a_{2,i}y + a_{3,i}z \\ &+ a_{4,i}x^2 + a_{5,i}y^2 + a_{6,i}z^2 \\ &+ a_{7,i}xy + a_{8,i}yz + a_{9,i}zx \end{aligned} \qquad (4.1)$$

where $a_{j,i}$, $j = 0, 1, 2, \ldots, 9$ are constants,

$\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$

x, y, z are rectangular coordinates (cm), and

$\vec{i}, \vec{j}, \vec{k}$ are unit vectors parallel to the x-, y-, and z- axes, respectively. The value of this equation, $u_i(\vec{r})$, is zero for points $\vec{r}$ on the surface.

These surfaces are described independently of the regions which define the material distributions to eliminate redundant input. Provision has been made in the FASTER program for recognizing more simple surfaces such as planes, cones, elliptical cylinders and ellipsoids. The equations for these simple surfaces are expanded by subroutine GEOMIN to obtain the coefficients of the general equation above. These special surfaces are shown in Figures 7, 8, and 9 and the equations are tabulated in Table 1 of the Section 9.3 input instructions.

4.2 SURFACE CALCULATIONS

Since surfaces are described independent of geometric regions, it is possible to define several quantities which are used in geometric calculations. For this discussion, the following are defined:

$\vec{r} = (x, y, z)$, the position vector of any point in space,

$\vec{\Omega} = (\alpha, \beta, \gamma)$, a unit vector defining the direction of a straight line, or ray, emanating from $\vec{r}$, where α, β, γ are direction cosines with respect to the x, y, and z axes, respectively,

$s, \; 0 \leq s < \infty$ the scalar distance from $\vec{r}$ along $\vec{\Omega}$

$\vec{r}' = \vec{r} + s\vec{\Omega}$ a point on the ray

$= (x + \alpha s, \; y + \beta s, z + \gamma s)$

Intersection of a Line and a Surface

The value of the quadric equation at $\vec{r}'$ is given by:

$$u_i(\vec{r}') = u_i(\vec{r}) + 2s\, v_i(\vec{r},\vec{\Omega}) + s^2 w_i(\vec{r},\vec{\Omega}) \qquad (4.2)$$

where $u_i(\vec{r})$ is given by equation 4.1 above and $v_i(\vec{r},\vec{\Omega})$ and $w_i(\vec{r},\vec{\Omega})$ are obtained by expanding this equation for $\vec{r}'$, and collecting the coefficients of s and s^2:

$$\begin{aligned} v_i(\vec{r},\vec{\Omega}) = & \tfrac{1}{2}\left[\alpha a_{1,i} + \beta a_{2,i} + \gamma a_{3,i}\right] \\ & + \alpha x a_{4,i} + \beta y a_{5,i} + \gamma z a_{6,i} \\ & + \tfrac{1}{2}\left[(\alpha y + \beta x) a_{7,i} + (\beta z + \gamma y) a_{8,i} + (\gamma x + \alpha z) a_{9,i}\right] \end{aligned} \qquad (4.3)$$

$$\begin{aligned} w_i(\vec{r},\vec{\Omega}) = & \alpha^2 a_{4,i} + \beta^2 a_{5,i} + \gamma^2 a_{6,i} \\ & + \alpha\beta a_{7,i} + \beta\gamma a_{8,i} + \gamma\alpha a_{9,i} \end{aligned} \qquad (4.4)$$

Intersections of the ray with the surface are obtained by requiring:

$u_i(\vec{r}') = 0$, i.e. this condition defines points on the surface using equation 4.2.

$$u_i(\vec{r}) + 2s_i v_i(\vec{r},\vec{\Omega}) + s_i^2 w_i(\vec{r},\vec{\Omega}) = 0 \qquad (4.5)$$

a) one intersection if $w_i(\vec{r},\vec{\Omega}) = 0$, $v_i(\vec{r},\vec{\Omega}) \neq 0$

$$s_i = -u_i(\vec{r}) / 2 v_i(\vec{r},\vec{\Omega}) \qquad (4.6)$$

b) two intersections if $w_i(\vec{r},\vec{\Omega}) \neq 0$, $v_i^2(\vec{r},\vec{\Omega}) > u_i(\vec{r})\, w_i(\vec{r},\vec{\Omega})$

$$s_i = \frac{-v_i(\vec{r},\vec{\Omega}) \pm \sqrt{v_i^2(\vec{r},\vec{\Omega}) - u_i(\vec{r})\, w_i(\vec{r},\vec{\Omega})}}{w_i(\vec{r},\vec{\Omega})} \qquad (4.7)$$

Note that the case of two equal intersections is not admitted since this is equivalent to no intersection. For all real intersections, the appropriate sign for multiple intersections is determined in the following manner:

a) The rate of change of $u_i(\vec{r}')$ with respect to distance at the intersection is given by differentiation of equation 4.2:

$$\left.\frac{\partial u_i(\vec{r}')}{\partial s}\right|_{s=s_i} = 2\left[v_i(\vec{r},\vec{\Omega}) + s_i\, w_i(\vec{r},\vec{\Omega})\right] \qquad (4.8)$$

Using the intersection equation 4.7 above:

$$\left.\frac{\partial u_i}{\partial s}(\vec{r}')\right|_{s=s_i} = 2\left[v_i(\vec{r},\vec{\Omega}) + w_i(\vec{r},\vec{\Omega})\left\{\frac{-v_i(\vec{r},\vec{\Omega}) \pm \sqrt{v_i^2(\vec{r},\vec{\Omega}) - u_i(\vec{r})\, w_i(\vec{r},\vec{\Omega})}}{w_i(\vec{r},\vec{\Omega})}\right\}\right]$$

$$= \pm 2\sqrt{v_i^2(\vec{r},\vec{\Omega}) - u_i(\vec{r})\, w_i(\vec{r},\vec{\Omega})} \qquad (4.9)$$

Thus this derivative must have the sign (+) used in the intersection equation.

b) It is noted that each surface defines two disjoint volumes, which, for the sake of a convention, are described as:

$$\left.\begin{array}{ll}\text{inner volume:} & u_i(\vec{r}\,') < 0 \\ \text{outer volume:} & u_i(\vec{r}\,') > 0\end{array}\right\} \qquad (4.10)$$

It follows that in crossing the surface from the inner volume, $u_i(\vec{r}\,')<0$, to the outer volume, $u_i(\vec{r}\,') > 0$, along any straight line, that the rate of change of the value of the surface equation $\partial u_i(\vec{r}\,')/\partial s$ is greater than zero at the intersection. Thus, if $u_i(\vec{r})< 0$ then the origin of the ray is inside the surface, and the first intersection with the surface is obtained from the quadratic equation 4.7 using the positive sign. A similar argument holds for crossing from the outside to the inside of the surface; i.e. if $u_i(\vec{r})>0$, then $\partial u_i(\vec{r}\,')/\partial s_i< 0$ at the first intersection implying the negative sign.

Surface Normal

The normal vector to the surface at the intersection, used in surface averaged flux calculations, is calculated by subroutine NORMAL as:

$$\vec{n} = \frac{\nabla u_i(\vec{r}\,')}{|\nabla u_i(\vec{r}\,')|} = (c^n_1, c^n_2, c^n_3) \qquad (4.11)$$

$$\text{where } \nabla = \vec{i}\frac{\partial}{\partial x'} + \vec{j}\frac{\partial}{\partial y'} + \vec{k}\frac{\partial}{\partial z'} \qquad (4.12)$$

$$\begin{aligned}\vec{n}\,|\Delta u_i(\vec{r}\,')| &= \left[a_{1,i} + 2x' a_{4,i} + a_{7,i}\, y' + a_{9,i}\, z'\right]\vec{i} \\ &+ \left[a_{2,i} + 2y' a_{5,i} + a_{8,i}\, z' + a_{7,i}\, x'\right]\vec{j} \\ &+ \left[a_{3,i} + 2z' a_{6,i} + a_{9,i}\, x' + a_{8,i}\, y'\right]\vec{k} \\ &= C_1\vec{i} + C_2\vec{j} + C_3\vec{k}\end{aligned} \qquad (4.13)$$

$$c^n_i = \frac{C_i}{\left[\sum_{i=1}^{3} C_i^2\right]^{1/2}}, \quad i = 1, 2, 3 \qquad (4.14)$$

4.3 REGIONS

Material Properties

The surfaces referred to in the previous section are used to describe the extent of geometric regions or zones having constant material properties. These properties are specified for each region, i, by a composition indicator m_i ($m_i < 0$ indicates that region i is void) and a separate hydrogen density ρ^h_i. The capability of specifying hydrogen densities by region simplifies the description of many problems, e.g., regions in a liquid hydrogen propellant tank. It is also helpful in describing hydrogen density variations in NERVA-type reactors where all other material properties are constant.

Source-in-Region

Additional regions may be required to correctly define the spatial extent of the fixed source volumes, i.e., there is an optional sampling technique in the FASTER program (described in Section 8.3) which requires that only one source be superimposed over a region and that the source cover the region. Alternatively, each source may cover more than one region.

This sampling technique is preferred since it requires much less a priori knowledge of the importance of various sources--point-kernel importance estimates are built in.

Region Boundaries

The geometric description of each region involves the listing of the surfaces which bound the region:

$k_{i,i}$, $i = 1, 2, \cdots$ where $k_{i,i}$ is the index of the surface forming the ith boundary of the region.

Also required are the components of an arbitrary point in the region:

$\vec{r}_i^{\,g} = x^g \vec{i} + y^g \vec{j} + z^g \vec{k}$ where (x^g, y^g, z^g) are specified.

Ambiguity Indices

The description of each region is completed in subroutine GEOMIN by computing an "ambiguity index" for each boundary surface. This ambiguity index indicates whether the region is inside or outside each of its boundaries. It also yields the sign to be used in calculating distances to quadratic or quadric boundaries. The ambiguity index is computed using equation 4.1:

$$\delta_{i,i} = - \frac{u_k(\vec{r}_i^{\,g})}{|u_k(\vec{r}_i^{\,g})|} \quad , \quad k = k_{i,i} \quad , \quad i = 1, 2, \cdots \tag{4.15}$$

Region Occupied by a Point

To ensure correct calculations, it is necessary that the ambiguity indices have the same sign for all points inside the region, i.e., a point $\vec{r}$ is in region i if, and only if $\delta_{i,i}\, u_{k_{ii}}(\vec{r}) < 0$ for all boundaries. The region index calculation for an arbitrary point is performed by the function subprogram LOCATE.

Possible Region Description Errors

Restrictions must be imposed on possible region shapes to ensure that all points in a region are always on the same side of each region boundary. For example, the single region

indicated in Figure 3A is unacceptable since there are points in the region which are both inside and outside boundaries A and B. The obvious solution is to use two regions to describe such geometric shapes.

It is sometimes necessary to introduce fictitious boundaries. Figure 3B shows a typical situation requiring these boundaries. Examination of the shaded and cross hatched regions reveals that they form two sections of a single region since ambiguity indices of the boundaries have the same values for both sections. This condition can cause trouble if only one section of the region is desired, even if other regions occupy the second section. The specification of the fictitious boundary eliminates the problem without otherwise affecting the geometric calculations. These fictitious boundaries must be included in the initial surface descriptions.

Geometry Consistency Check

A rather simple check for correct geometric description involves a calculation of the region(s) occupied by each point $\vec{r}_i^{\,g}$. If the point $\vec{r}_i^{\,g}$ is in any region other than region i, the geometry representation is incorrect. This geometry check is performed by the GEOMIN subroutine using the LOCATE function subprogram.

4.4 RAY TRACING

The procedure for ray tracing is similar to that discussed above for the intersection of a ray with individual surfaces. The cumulative path lengths through each region along a ray are computed at a single pass, in the order traversed, by subroutine PATH. As indicated before, the ray tracing calculations are related to the surfaces.

$$s_{l+1} = \min_{i = 1, 2, \cdots} \left\{ \begin{array}{l} s_{i,i} \text{ such that } s_{i,i} \geq s_l \text{ , and} \\ \text{if } s_{i,i} = s_l \text{ , } \delta_{i,i}\, \partial u_k(\vec{r} + s_l \vec{\Omega})/\partial s > 0 \end{array} \right\} \tag{4.16}$$

s_l = distance up to the region, and

$s_{i,i}$ = is obtained from equation 4.6 or 4.7 for surface $k = k_{i,i}$ using the sign of $\delta_{i,i}$ for non-planar surfaces.

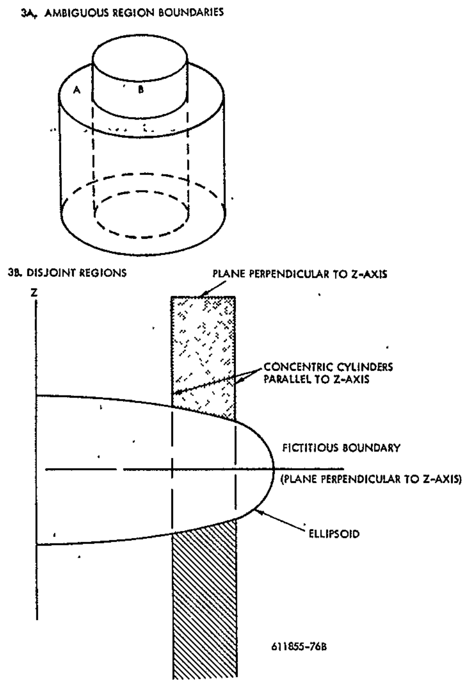


Figure 3. Problems in Region Descriptions

Astronuclear Laboratory

The surface crossed in leaving the region is that giving the minimum distance.

If the new total distance s_{l+1} exceeds a specified maximum, e.g., the distance between two points, the distance through the region is adjusted and the ray tracing terminates. If not, the next region entered by the ray is computed by requiring that:

a) the next region have the surface just crossed as a boundary,

b) the next region be on the other side of this surface (opposite signs on ambiguity indices), and

c) for all other boundaries

$$u_{k'}(\vec{r} + s_{l+1}\vec{\Omega})\ \delta_{j',i'} \le 0, \quad k' = k_{j',i'}, \quad j' = 1, 2, \ldots \qquad (4.17)$$

which is evaluated using equation 4.2.

The index of the region entered in crossing the above boundary is saved as the most probable next region for subsequent ray tracings. If no region is accepted, the exterior of the geometry is assumed and an indicator set. After several subsequent failures, testing for this boundary is never performed.

The constants $u_k(\vec{r})$, $v_k(\vec{r}, \vec{\Omega})$, and $w_k(\vec{r}, \vec{\Omega})$ and the intersections are computed only once during a given ray tracing. The current status of calculations for the k th surface is indicated by n_k where

a) $n_k = 0$ if the k th surface has not been involved as yet, in the ray tracing,

b) $n_k = 1$ if the constants $u_k(\vec{r})$, $v_k(\vec{r}, \vec{\Omega})$, $w_k(\vec{r}, \vec{\Omega})$ have been computed,

c) $n_k = 2$ if the constants and the intersections have both been calculated and at least one intersection is at a distance greater than the current cumulative distance s_l, and

d) $n_k = 3$ if the constants and intersections have been computed but the intersections need no longer be considered e.g., if both are less than s_l, imaginary, etc.

SECTION

5.0 FIXED SOURCES

A number of numerical techniques can be used for describing fixed source distributions. Those employed in the FASTER program incorporate the assumption of separable variables. These techniques are general enough to permit the description of a variety of real source distributions and the distributions used in generating basic data.

5.1 SPATIAL AND ANGULAR VARIABLES

The FASTER program will handle multiple sources in rectangular, cylindrical or spherical geometries. The geometry for each source is superimposed over the various geometric regions. The source geometries for each of the multiple sources need not be the same. In all geometries there are three spatial variables (v_1, v_2, v_3) and two angular variables(v_4, v_5). The relationships between these spatial and angular variables are shown in Figure 10 in the Section 9.4 input instructions.

Rectangular Geometry

The most simple geometry is that used for describing rectangular source volumes. The spatial variables (v_1, v_2, v_3) are the rectangular coordinates (x, y, z). The angular variables (v_4, v_5) are the azimuthal angle θ measured from the x-axis and the cosine of the polar angle, μ, measured from the z-axis.

Cylindrical Geometry

The next allowed geometry, usually used in describing reactor sources, involves cylinders parallel to the z-axis. The spatial variables are:

$v_1 = r = \sqrt{x^2 + y^2}$, the radius

$v_2 = \theta = \tan^{-1}(y/x)$, the azimuthal angle measured from the x-axis

$v_3 = z$, the axial coordinate

The angular variables are measured in a coordinate system which rotates with the radius vector. The variables are:

$v_4 = \theta'$, the azimuthal angle measured from θ.

$v_5 = \mu'$, the cosine of the polar angle measured from the z-axis.

This simplifies the description of angular sources on the surfaces of a cylindrical reactor.

Spherical Geometry

The final geometry, useful in describing sources such as capture gammas in the hemispherical bottom of a liquid hydrogen propellant tank, involves the spatial variables:

$v_1 = \rho = \sqrt{x^2 + y^2 + z^2}$, the spherical radius

$v_2 = \theta = \tan^{-1}(y/x)$ the azimuthal angle measured from the x-axis.

$v_3 = \mu = \frac{z}{\rho}$, the cosine of the polar angle measured from the z-axis.

The coordinate system used for the angular variables rotates with the spherical radius vector:

$v_4 = \theta'$, the azimuthal angle measured as shown in Figure 10

$v_5 = \mu'$, the cosine of the polar angle measured from the spherical radius vector.

Source Translations

In addition, each source is given a translation vector $\vec{r}_t = (x_t, y_t, z_t)$ from the origin of the geometry coordinate system. Thus, the coordinates of the source points, expressed in the geometry coordinate system, are

$$\vec{r} = (x + x_t, y + y_t, z + z_t)$$

5.2 SPATIAL AND ANGULAR DISTRIBUTIONS

Each of the distributions for the spatial and angular variables is described separately by tabulating relative distributions

$$(v_{k,i}, f_{k,i}), \quad k = 1, 2, \cdots$$

where $v_{k,i}$ is the kth value of the ith variable, and $f_{k,i}$ is the relative distribution at $v_{k,i}$.

Each variable v_i may take on only one value $v_i = v_{1,i}$ i.e., $f(v_i) = \delta(v_i - v_{1,i})$. If more than one point is needed, then $f(v_i)$ is assumed to be continuous.

The continuous distributions are normalized in subroutine SOURCE by integrating a linear interpolation formula and requiring that:

$$\sum_{k=1} \int_{v_{k,i}}^{v_{k+1,i}} \frac{(v_{k+1,i} - v_i) f_{k,i} + (v_i - v_{k,i}) f_{k+1,i}}{v_{k+1,i} - v_{k,i}} v_i^n dv_i = 1 \qquad (5.1)$$

where $n = 0$ except for the radial distributions of cylindrical and spherical sources where $n = 1$ or 2, respectively.

The final representation of the spatial and angular distributions is the product of the individual distributions for the five source variables:

$$p(\vec{r}, \vec{\Omega}) = \prod_{i=1}^{5} f(v_i) \qquad (5.2)$$

5.3 SOURCE SPECTRA

Particle energies generally decrease with the increase in the order of scattering and the FASTER program requires the same order in energies, i.e., a series of energy groups are defined with group 1 containing particles with the maximum energy. The same group structure is used for the source spectra and later for the cross sections. These energy groups are defined in subroutine SOURCE by:

$$\text{Group } j : \quad E_j \geq E\ (\text{Mev}) \geq E_{j+1} \qquad j = 1, 2, \cdots \tag{5.3}$$

Some relaxation is allowed on describing the source spectra in that an arbitrary group structure, with energy group boundaries of decreasing energy, can be used. Various quantities are accepted as input. They are all reduced, however, to one form; a differential number spectrum:

$$\eta_k \left(\frac{\text{particles}}{\text{Mev} \cdot \text{sec}}\right) \quad \text{at energy} \quad E_k \qquad k = 1, 2, \cdots$$

where the E_k's define the input energy group boundaries.

This spectrum is then integrated into the group structure for the problem using a linear interpolation formula for the energy variation. The final spectrum is expressed as the number of particles in each group and the average energy of these particles:

$$n_j^o = \sum_{\substack{k=1 \\ E_{jk}^l < E_{jk}^h}} \int_{E_{jk}^l}^{E_{jk}^h} \eta_k(E)\, dE \qquad \text{(particles in group } j\text{)} \tag{5.4}$$

$$\overline{E_j^o} = \frac{1}{n_j^o} \sum_{\substack{k=1 \\ E_{jk}^l < E_{jk}^h}} \int_{E_{jk}^l}^{E_{jk}^h} \eta_k(E)\, E\, dE \qquad \text{(average energy of the particles in group } j\text{)} \tag{5.5}$$

where

$$\eta_k(E) = \frac{(E - E_{k+1})\,\eta_k + (E_k - E)\,\eta_{k+1}}{E_k - E_{k+1}} \tag{5.6}$$

$$\left.\begin{aligned} E_{jk}^l &= \max(E_{j+1}, E_{k+1}) \\ E_{jk}^h &= \min(E_j, E_k) \end{aligned}\right\} \tag{5.7}$$

The groupwise number spectrum n_j^o, $j = 1, 2, \cdots$, is then normalized to an input total source strength.

5.4 FIXED SOURCE ACQUISITION

Volume Sources

The definition of neutron and photon source distributions for reactor configurations can be a time-consuming task. For geometries where the discrete ordinate (S_n) methods are applicable, source distributions can be obtained both efficiently and economically through their use. In particular, the coupled ODD-K - NAGS system (References 11, 12) can provide the relative distributions and spectra in the form required by the FASTER program. Included are separable radial and axial distributions and spectra for all reactor regions.

Angular Surface Fluxes

The design of nuclear rocket engines is such that given the internal reactor arrangements it is possible to define the reactor leakage within an error involving external reactivity effects and that inherent in the calculational method. This same leakage can then be applied to a variety of external problems such as individual external components or a liquid hydrogen propellant tank.

Assuming a detailed internal calculation, it is possible to numerically integrate the equation for the unperturbed angular flux at an arbitrary point in space outside the reactor:

$$\phi^U(\vec{r},\vec{\Omega},E) = \int_0^\infty S(\vec{r}-s\vec{\Omega},\vec{\Omega},E)\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right] ds$$

$$= \int_{s_0}^{s_1} S(\vec{r}-s\vec{\Omega},\vec{\Omega},E)\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right] ds$$

$$\phi^U(\vec{r},\vec{\Omega},E) = \int_0^{s_1-s_0} S(\vec{r}-s_0\vec{\Omega}-t\vec{\Omega},\vec{\Omega},E)\exp\left[-\int_0^t \Sigma^t(\vec{r}-s_0\vec{\Omega}-t'\vec{\Omega},E)\,dt'\right] dt$$

$$= \phi(\vec{r}-s_0\vec{\Omega},\vec{\Omega},E) \qquad (5.8)$$

where

$S(\vec{r}',\vec{\Omega},E)$	is the total differential source density in the reactor, i.e., including scattering.
s_0	is the distance to the reactor surface from $\vec{r}$
s_1	is the distance through the reactor from $\vec{r}$
$\phi(\vec{r}-s_0\vec{\Omega},\vec{\Omega},E)$	is the angular flux at the reactor surface, i.e., at $\vec{r}-s_0\vec{\Omega}$

(For numerical integrations, the equation is sometimes transformed to an area integration over the reactor surface.)

A more general use of the angular leakage fluxes is as a surface source (or a "thin" volume source). The requisite data reduction of ODD-K angular leakage fluxes, both neutron and photon, is performed by the DAFT program.(13) Included in this reduction are averages over arbitrary mesh points, groups, etc.

The external environment is then given by a set of equations analogous to those in Section 2.2, where:

$$\phi_0(\vec{r},\vec{\Omega},E) = \phi(\vec{r}-s_0\vec{\Omega},\vec{\Omega},E)\exp\left[-\int_0^{s_0} \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right] \qquad (5.9)$$

These reduced angular fluxes thus fit into the framework provided in the FASTER program for describing fixed sources, since the equation for the external uncollided flux is identical to that obtained for a surface source.

5.5 SOURCE EVALUATION

The source was defined as a function of the variables $v_1, v_2, \ldots, v_5$. For the order of scatter flux calculations, the source must be evaluated for a specified point $\vec{r}_0$ and direction $\vec{\Omega}$. This evaluation, when divided by the value $p_0^*(\vec{r}_0)$ of the sampling function used in obtaining the position vector $\vec{r}_0$, yields the energy dependent angular point source $W_0^s(\vec{\Omega},E)$. The source evaluation is performed by subroutine SZERO and involves several variable transformations to obtain the source variables equivalent to $\vec{r}_0$ and $\vec{\Omega}$:

a) define the source centered position vector,
$$\vec{r} = \vec{r}_0 - \vec{r}_t = (x, y, z)$$
b) calculate the spatial variables (v_1, v_2, v_3) from the rectangular coordinates using the transformation equations for the source geometry,
c) calculate the direction vector rotation matrix, $R_{i,j}$, (See Figure 4.),
d) calculate the direction cosines in the rotated coordinate system for which the angular distribution of the source is defined

$$c_j' = \sum_{i=1}^{3} c_i R_{ij}, \quad j = 1, 2, 3 \quad \text{where } \vec{\Omega} = c_1\vec{i} + c_2\vec{j} + c_3\vec{k}$$

e) calculate the angular variables v_4 and v_5,

$$v_4 = \tan^{-1} (c'_2/c'_1) \qquad v_5 = c'_3$$

A linear interpolation of the source variable distributions yields:

$$\left.\begin{aligned} W_i^s &= \left[\frac{\prod_{i=1}^{5} f(v_i)}{P_o^*(\vec{r}_o)}\right] n_i^o, \\ \overline{E}_i^s &= E_i^o \end{aligned}\right\} \quad i = 1, 2, \cdots \tag{5.10}$$

where W_i^s is the number of particles in group i and $\overline{E}_i^s$ is the average energy of the particles in group i. The average group energy $\overline{E}_i^s$ is just $\overline{E}_i^o$ as calculated during spectrum normalization.

Thus the general equation 2.23; for $W_o^s(\vec{\Omega}, E)$ is reduced to a groupwise representation:

$$W_i^s = \int_{E_{i+1}}^{E_i} W_o^s(\vec{\Omega}, E)\, dE \tag{5.11}$$

$$\overline{E}_i^s = \frac{1}{W_i^s} \int_{E_{i+1}}^{E_i} W_o^s(\vec{\Omega}, E)\, E\, dE$$

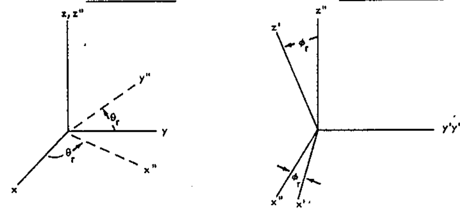
The angular dependence has been suppressed since, in the numerical calculations, only one discrete direction is considered at a time. Moreover, the order-of-scatter subscript has also been suppressed since it is used only as a counter in the numerical calculations.

The same representation of the point angular sources is used for higher order-of-scatters i.e. for $W_k^s(\vec{\Omega}, E)$. The equations are discussed in the next section.

$\vec{\Omega}$ = unit direction vector in geometry coordinate system
$= c_1\vec{i} + c_2\vec{j} + c_3\vec{k}$

$\vec{\Omega}'$ = unit direction vector in rotated coordinate system
$= c_1'\vec{i}' + c_2'\vec{j}' + c_3'\vec{k}'$

FIRST ROTATION — SECOND ROTATION



$$c_i = \sum_{j=1}^{3} c_j' R_{j,i} \qquad i = 1,2,3 \qquad\qquad c_i' = \sum_{j=1}^{3} R_{ij} c_j \qquad i = 1,2,3$$

$R_{1,1} = \cos\theta_r \cos\phi_r$	$R_{1,2} = \sin\theta_r \cos\phi_r$	$R_{1,3} = -\sin\phi_r$
$R_{2,1} = -\sin\theta_r$	$R_{2,2} = \cos\theta_r$	$R_{2,3} = 0$
$R_{3,1} = \cos\theta_r \sin\phi_r$	$R_{3,2} = \sin\theta_r \sin\phi_r$	$R_{3,3} = \cos\phi_r$

SOURCE GEOMETRY ROTATION ANGLES

ANGLE	RECTANGULAR	CYLINDRICAL	SPHERICAL
θ_r (azimuthal)	0	$\tan^{-1}(y/x)$	$\tan^{-1}(y, x)$
ϕ_r (polar)	0	0	$\tan^{-1}\left(z/\sqrt{x^2+y^2+z^2}\right)$

Figure 4. Direction Vector Rotations

SECTION

6.0 TRANSPORT AND SCATTERING KERNELS

The equations used in the FASTEP program for defining the attenuating and scattering properties of homogeneous material compositions are described below. The general procedure involves a calculation of macroscopic cross sections using input microscopic data. The two most common sets of units for microscopic data and compositions are both permitted in FASTER:

u_x = 0 for microscopic cross sections in barns/atom

= 1 for microscopic cross sections in cm^2/gm

u_c = 0 for compositions in 10^{24} atoms/cm^3

= 1 for compositions in gm/cm^3

i.e., the composition and cross section units can be mixed.

6.1 PHOTON CROSS SECTIONS

The equations used for photon cross sections are discussed in this section. Photon cross sections are defined at the boundaries of the energy groups defined by equation 5.3. The requisite data for the ith element is:

A_i the atomic mass (a.m.u.) of the element

Z_i the atomic number of the element

$\rho_{m,i}$ the density of the element in composite material m with units according to u_c, and

$\sigma^t_{j,i}$ the microscopic total cross section for energy level j with units according to u_x.

The total cross section, $\Sigma^t_{j,m}$, is computed by subroutine INSECT for each energy level of each composite material by a summation over the element data:

$$\Sigma^t_{j,m} = \sum_{i=1} \left(\frac{0.6025}{A_i}\right)^{(u_c - u_x)} \rho_{m,i}\ \sigma^t_{j,i} \qquad (cm^{-1}) \qquad (6.1)$$

The total electron density is calculated by:

$$n_m^e = \sum_{i=1} \left(\frac{0.6025}{A_i}\right)^{\upsilon_c} \rho_{m,i}\, Z_i \left(\frac{10^{24}\ \text{electrons}}{\text{cm}^3}\right) \qquad (6.2)$$

where $(0.6025/A_i)^{\upsilon_c}$ converts compositions to 10^{24} atoms/cm^3, and $(0.6025/A_i)^{-\upsilon_x}$ converts cross sections to barns/atom.

The energy absorption coefficient by energy level is computed by element and material. The absorbtion coefficients can then be used as a flux-to-heating conversion factor under conditions noted in Section 9.5 of the input instructions. The equation for the microscopic energy absorption coefficients assumes all interactions, except Compton scattering, are absorptions (Reference 4, pg. 159):

$$\mu^a_{i,i} = \sigma^t_{i,i} - \frac{1.995}{8}\left(\frac{0.6025}{A_i}\right)^{\upsilon_x}\left[\frac{\ln(1+2\eta)}{\eta^3} + \frac{2(1+\eta)(2\eta^2-2\eta-1)}{\eta^2(1+2\eta)} + \frac{8\eta^2}{3(1+2\eta)^3}\right] \qquad (6.3)$$

with units according to υ_x and where $\eta = E_i/0.511$.

6.2 PHOTON TRANSPORT

The material attenuation kernel is written for the ith energy group as:

$$K^t_i(\vec r, \vec r') = \exp\left[-\int_0^s \Sigma^t(\vec r + s'\vec\Omega, \overline{E^s_i})\,ds'\right],\quad s = |\vec r - \vec r'|,\ \vec\Omega = (\vec r - \vec r')/s \qquad (6.4)$$

This kernel is evaluated using $\overline{E^s_i}$, the average group energy of the multigroup representation of the angular point sources defined in equation 5.10. The assumption of transport without change in the average group energy simplifies this kernel.

The evaluation of this kernel uses a group averaged cross section obtained by a linear, energy interpolation of the cross sections at the group boundaries. Since each geometric region has constant material properties,

$$K_i^t(\vec r, \vec r') = \exp\left[-\sum_{l=1} \Sigma_i^v\, \Delta s_l\right] \qquad (6.5)$$

where Δs_l is the distance in the lth region traversed from $\vec r$ to $\vec r'$, and

Σ_i^v is the average total cross section for the region and group i

This total cross section can be composed of two parts; one representing the composite material and another representing the hydrogen in the region as discussed in Section 4.3.

This photon attenuation kernel is used to calculate fluxes, and, in particular, to define the point monodirectional flux components which are used to represent the next order-of-scatter point source; i.e., equation 2.27

$$W_i^\phi = \frac{W_i^s\, K_i^t(\vec r_{k-1}, \vec r_k)}{|\vec r_k - \vec r_{k-1}|^2\, p_k^*(\vec r_k)},\quad \overline{E_i^\phi} = \overline{E_i^s} \qquad (6.6)$$

where $p_k^*(\vec r_k)$ is the value of the sampling function used to obtain $\vec r_k$.

This is merely a groupwise representation of equation 2.27, where the average energy of the particles in each group at the scattering point is assumed equal to the average energy at the previous source point:

$$W_i^\phi = \int_{E_{i+1}}^{E_i} W_k^\phi(E)\,dE$$

$$\overline{E_i^\phi} = \frac{1}{W_i^\phi}\int_{E_{i+1}}^{E_i} W_k^\phi(E)\,E\,dE$$

$$\approx \overline{E_i^s}$$

6.3 PHOTON SCATTERING

The photon scattering calculations performed in FASTER by subroutine SINGLE, use the Klein-Nishina equation for Compton scattering. All scattering is assumed to occur at *the average group energy* $\overline{E_j^{\phi}}$ *so that the ratio of the energies before and after scattering* for group j is:

$$R_j = \frac{0.511}{0.511 + \overline{E_j^{\phi}}(1-\mu)} \qquad (6.7)$$

where $\mu = \vec{\Omega}_k \cdot \vec{\Omega}$, the cosine of the scattering angle, and

$$\vec{\Omega}_k = (\vec{r}_k - \vec{r}_{k-1})/|\vec{r}_k - \vec{r}_{k-1}| \qquad (6.8)$$

The scattered point source component--the scattered particles due to particles originally in group j--is then obtained from the Klein-Nishina equation:

$$\Delta W_j^s = W_j^{\phi} \frac{0.49875}{4\pi} N_e \; R_j^2 \left[R_j + \frac{1}{R_j} - 1 + \mu^2\right] \qquad (6.9)$$

where N_e is the total electron density for the region in which the scattering occurs--including that due to the separate hydrogen.

The scattered contributions are grouped, according to the average scattered energies, to yield the final representation of the scattered point source:

$$\left.\begin{aligned} W_{j'}^s &= \sum_{j \in J'} \Delta W_j^s \\ \overline{E_{j'}^s} &= \frac{1}{W_{j'}^s} \sum_{j \in J'} \Delta W_j^s R_j \overline{E_j^{\phi}} \end{aligned}\right\} \; j \in J' \quad \text{if} \quad E_{j'} > R_j \overline{E_j^{\phi}} \geq E_{j'+1} \qquad (6.10)$$

where $R_j\overline{E_j^{\phi}}$ is the average energy after scattering of particles originally in group j, and J' is the set of initial group indices j, for which this scattered energy is within the boundaries of group j'.

The angular point source at the scattering point is evaluated only for discrete directions $\vec{\Omega}$. Therefore, the angular dependence has been suppressed in this development.

6.4 NEUTRON CROSS SECTIONS

Neutron transport and scattering calculations utilize group averaged cross sections only. These can be obtained from various tabulations, e.g., Reference 14, which also discusses the averaging techniques.

The microscopic data for the ith element is supplied to the FASTER program by energy group j as:

$\sigma^t_{j,i}$ — *the average total cross section for element i and group j*

$\sigma^l_{j \to k, i}$ — the lth Legendre expansion coefficient for element i of the differential elastic scattering cross section for transfer from group j to group k,

$\sigma^{ne}_{j \to k, i}$, — an isotropic, weighted, non-elastic transfer cross section

$$\sigma^{ne}_{j\to k,i} = \sigma^{inelastic}_{j\to k,i} + 2\cdot\sigma^{(n-2n)}_{j\to k,i} + \ldots$$

The elastic scattering coefficients are assumed to contain the (2l + 1) factor associated with the Legendre series expansion.

The INSECT subroutine will transport correct these neutron cross sections, or it will remove the transport correction under conditions noted in the input instructions in Section 9.5. In these cross section manipulations, the first two Legendre expansion coefficients of the group-averaged elastic scattering cross sections are computed first:

$$\sigma^{1}_{j,i} = \sum_{k} \sigma^{1}_{j\rightarrow k,i} \qquad (6.12)$$

where $\sigma^{o}_{j,i}$ is the average total elastic scattering cross section for the jth group of the ith element

$\sigma^{1}_{j,i} = 3\bar{\mu}_{j,i}\,\sigma^{o}_{j,i}$

$\bar{\mu}_{j,i}$ is the average cosine of the scattering angle in the laboratory coordinate system for group j of the ith element.

The correction of the group averaged total cross section yields the group averaged transport cross section $\sigma^{tr}_{j,i}$:

$$\sigma^{tr}_{j,i} = \sigma^{t}_{j,i} - \bar{\mu}_{j,i}\,\sigma^{o}_{j,i} = \sigma^{t}_{j,i} - \sigma^{1}_{j,i}/3 \qquad (6.13)$$

The removal of this correction requires the equation:

$$\sigma^{t}_{j,i} = \sigma^{tr}_{j,i} + \sigma^{1}_{j,i}/3$$

If the total cross section is transport corrected, the corresponding correction to the group averaged elastic scattering cross section is applied to the in-group term of the isotropic Legendre expansion coefficient, $\sigma^{o}_{j\rightarrow j,i}$ only:

$$\sigma^{o,tr}_{j\rightarrow j,i} \doteq \sigma^{o}_{j\rightarrow j,i} - \sigma^{1}_{j,i}/3 \qquad (6.14)$$

Special attention is given to hydrogen since its presence in large amounts is not adequately represented by the above group-to-group transfer cross sections. Hydrogen cross sections are supplied as group averaged totals σ^{h}_{j}. The scattering cross section is then assumed to equal the total cross section and the angular dependence of the scattering is treated correctly as described in Section 6.5.

Macroscopic cross sections, by composite material, are computed by energy group in a manner similar to that for photon cross sections. This includes the total cross section $\Sigma^{t}_{j,m}$ and the scattering cross sections $\Sigma^{l}_{j\rightarrow k,m}$ and $\Sigma^{ne}_{j\rightarrow k,m}$ or the equivalent transport corrected cross sections.

Kinetic heating responses are computed by INSECT for each element and combined by composite material for use as heating conversion factors. Computed as average fractional energy loss cross sections, the groupwise equations are:

$$\sigma\left[1-\overline{\left(\frac{E_{out}}{E_{in}}\right)}\right]_{j,i} = \frac{2A_i}{(A_i+1)^2}\,\sigma^{o}_{j,i}\left(1-\bar{\mu}^{c.m}_{j,i}\right) \qquad (6.15)$$

where E_{in} is the energy of neutrons going into the elastic collision,

E_{out} is the energy of neutrons coming out of this collision,

$1-\overline{\left(\frac{E_{out}}{E_{in}}\right)}$ is the average fractional energy loss

$\sigma^{o}_{j,i}$ is the total elastic scattering cross section as given by equation 6.12 for heavy elements ($A_i > 1$)

$= \sigma^{h}_{j}$ for hydrogen

$\bar{\mu}^{c.m}_{j,i}$ is the average scattering angle cosine in the center-of-mass coordinate system

$\bar{\mu}^{c.m}_{j,i} = 0$ for hydrogen

$$\bar{\mu}^{c.m}_{j,i} \approx \frac{1}{A_i}\left[\bar{\mu}^{2}_{j,i} - 1 + \bar{\mu}_{j,i}\sqrt{A_i^{2} + \bar{\mu}^{2}_{j,i} - 1}\right] \text{ for heavy elements} \qquad (6.16)$$

$\bar{\mu}_{j,i}$ is the average scattering angle cosine in the laboratory coordinate system.

6.5 NEUTRON ATTENUATION AND SCATTERING

The neutron attenuation kernel $K_i^t(\vec{r}, \vec{r}')$ is handled in the same manner as the photon attenuation kernel. The only difference is that there is no energy interpolation required to define the average group cross sections.

The neutron source from scattering is developed in two parts. The first calculation yields the neutron source contribution from heavy element scattering:

$$\Delta W_{i'}^{H} = \sum_{i=1} W_i^{\phi} \frac{1}{4\pi} \left[\Sigma_{i \mapsto i', m}^{ne} + \sum_{l=0} \Sigma_{i \mapsto i', m}^{l} \; P_l(\mu) \right] \tag{6.17}$$

where m denotes the material at the scattering point

μ is the cosine of the scattering angle, $\mu = \vec{\Omega}_k \cdot \vec{\Omega}$

$P_l(\mu)$ is the lth Legendre polynomial, i.e., from Reference 16, pg. 308:

$$\left.\begin{aligned} P_o(\mu) &= 1 \\ P_1(\mu) &= \mu \\ P_l(\mu) &= \frac{1}{l}\left[(2l-1)\,\mu P_{l-1}(\mu) - (l-1)\,P_{l-2}(\mu)\right] \end{aligned}\right\} \tag{6.18}$$

The neutron source for scattering from hydrogen is obtained only if $\mu > o$, i.e., scattering in the laboratory coordinate system is restricted to angles < 90 degrees. Then the scattered source contribution is computed as:

$$\Delta W_{i'}^{h} = 4\mu\rho^h \sum_{i \in J'} W_i^{\phi} \frac{\sigma_i^h}{4\pi} \tag{6.19}$$

where $\mu^{c.m} = 2\mu^2 - 1$, $d\mu^{c.m}/d\mu = 4\mu$, $E_{out}/E_{in} = \frac{1+\mu^{c.m}}{2} = \mu^2$,

$i \in J'$ if $E_{i'} > \mu^2 \; \overline{E_i^{\phi}} \geq E_{i'+1}$ i.e., if the scattered energy $\mu^2 \overline{E_i^{\phi}}$ is within the boundaries of group i'; and ρ^h is the local hydrogen density.

These two components are then combined to yield the final value of the neutron scattered point source:

$$W_{i'}^{s} = \Delta W_{i'}^{H} + \Delta W_{i'}^{h}$$

$$\overline{E_{i'}^{s}} = \frac{1}{W_i^s} \left[\overline{E_{i'}^{n}} \; \Delta W_{i'}^{H} + 4\mu\rho^h \sum_{i \in J'} W_i^{\phi} \frac{\sigma_i^h}{4\pi} \mu^2 \; \overline{E_i^{\phi}} \right] \tag{6.20}$$

where the $\overline{E_{i'}^{n}}$, $i' = 1, 2, \ldots$, are input average scattered energies for neutron scattering from heavy elements.

SECTION

7.0 FLUX ESTIMATION

The FASTER program computes multigroup neutron or photon flux components for arbitrarily located point, surface and/or volume detectors. The surface averaged fluxes are obtained for specified sections of surfaces which form boundaries of regions. Thus there may be two equivalent definitions of each surface detector, i.e., if two regions have the desired section of the surface as a common boundary. Volume averaged fluxes are obtained for specified regions of the geometry using the two special equations 2.55 and 2.56 for void regions and constant material density regions respectively.

7.1 ANGULAR FLUX CONTRIBUTIONS

The random sampling technique discussed in Section 2.7 is used in integrating the angular dependence of the point sources to obtain surface and volume averaged flux components. Therefore, the individual contributions for all detector types have a similar form involving discrete directions. A full set of indices will be used in the following summary of flux estimation equations, i.e., the contribution to the <u>jth</u> energy group from the <u>kth</u> inner iteration of the <u>ith</u> outer iteration is given by:

a) point detector at $\vec{r}$

$$\Delta\phi^{*}_{ijk}\ (\vec{\Omega}) = \left\{ \frac{W_j^s\ K_j^t\ (\vec{r}_{ik}, \vec{r})}{\left| r - r_{ik} \right|^2} \right\} \delta(\vec{\Omega} - \vec{\Omega}_o) \tag{7.1}$$

$$\vec{\Omega}_o = (\vec{r} - \vec{r}_{ik}) / |\vec{r} - \vec{r}_{ik}|$$

b) surface detector at a distance s from $\vec{r}_{ik}$ along $\vec{\Omega}_l$

$$\Delta\phi^{*}_{ijk}\ (\vec{\Omega}) = \left\{ \frac{W_{ijk}^s\ K^t\ (\vec{r}_{ik}, \vec{r}_{ik} + s\vec{\Omega}_l)}{A \cdot L \cdot |\vec{\Omega}_l \cdot \vec{n}|\ q^*(\vec{\Omega}_l)} \right\} \delta(\Omega - \Omega_o) \tag{7.2}$$

$$\vec{\Omega}_o = \vec{\Omega}_l$$

c) void volume detector at a distance s from $\vec{r}_{ik}$ along $\vec{\Omega}_l$

$$\Delta\phi^*_{ijk}(\vec{\Omega}) = \left\{ \frac{W^s_i \; K^t_i \; (\vec{r}_{ik}, \vec{r}_{ik} + s\Omega_l)\,\Delta s}{V \cdot L \cdot q^*(\vec{\Omega}_l)} \right\} \delta(\vec{\Omega} - \vec{\Omega}_o) \qquad (7.3)$$

$$\vec{\Omega}_o = \vec{\Omega}_l$$

d) non-void volume detector at a distance s from $\vec{r}_{ik}$ along $\vec{\Omega}_l$

$$\Delta\phi^*_{ijk}(\vec{\Omega}) = \left\{ \frac{W^s_i \; K^t_i \; (\vec{r}_{ik}, \vec{r}_{ik} + s\vec{\Omega}_l)}{V \cdot L \cdot q^*(\Omega_l)} \cdot \frac{1 - \exp\left[-\Delta s \;\; \Sigma^v_i\right]}{\Sigma^v_i} \right\} \delta(\vec{\Omega} - \vec{\Omega}_o) \qquad (7.4)$$

$$\vec{\Omega}_o = \vec{\Omega}_l$$

or in general:

$$\Delta\phi^*_{ijk}(\vec{\Omega}) = \Delta\phi_{ijk} \;\; \delta(\vec{\Omega} - \vec{\Omega}_o) \qquad (7.5)$$

where $\Delta\phi_{ijk}$ is the quantity in brackets { } above. The angular dependence implied by the Dirac delta function is only a formility for use in defining angular moments of the flux as discussed in Appendix B. This multigroup representation of the point angular source is obtained, of course, for the discrete direction $\vec{\Omega}_o$. Surface and volume detectors may receive more than one flux contribution. This may occur if more than one discrete direction, $\vec{\Omega}_l$, ($L > 1$) is used in the angular integration of the point angular sources. It will occur if the detectors are intercepted more than once for each discrete direction.

7.2 SCALAR FLUXES AND COMPONENTS

The full set of indices, outer iteration i, inner iteration k, and energy group j, was introduced to simplify the discussion of the final equations for the various flux components. Unless a flux component is used in a subsequent equation, there will be no symbol introduced on the left side of the equation for this component.

The FASTER program actually includes a group collapse, if desired, to a coarser group structure for the flux edits. This will be ignored in the equations below since it involves nothing more than an inner summation. Thus, a summation over energy groups before any other summation is implicit. Also implicit is a summation over multiple intersections and/or discrete directions for surfaces and volumes.

Scalar Number Flux (Sample Mean)

The total scalar flux in group j is obtained by an angular integration:

$$\phi^o_j = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0} \iint_{4\pi} \Delta\phi_{ijk}\,\delta(\vec{\Omega} - \vec{\Omega}_o)\, d\Omega \qquad (7.6)$$

$$= \frac{1}{n} \sum_{i=1} \sum_{k=0} \Delta\phi_{ijk}$$

Other quantities are obtained in a similar manner. The index j is reserved for the energy dependence in all of the equations below.

Sample Variance of Scalar Flux

$$V^2_j = \frac{1}{n-1} \left[\sum_{i=1}^{n} \left(\sum_{k=0} \Delta\phi_{ijk} \right)^2 - n\,(\phi^o_j)^2 \right] \qquad (7.7)$$

Relative Error

$$E_j = \frac{\sqrt{V^2_j / n}}{\phi^o_j} \qquad (7.8)$$

Differential Number Flux

$$\phi_i^o / (E_i - E_{i+1}) \qquad (7.9)$$

Cumulative Number Flux $(E \geq E_{i+1})$

$$\sum_{i'=1}^{i} \phi_{i'}^o \qquad (7.10)$$

Energy Flux

$$I_i^o = \frac{1}{n} \sum_{i=1}^{n} \overline{E_{ijk}^s} \; \Delta\phi_{ijk} \qquad (7.11)$$

Average Group Energy

$$\overline{E}_i = I_i^o / \phi_i^o \qquad (7.12)$$

Differential Energy Flux

$$I_i^o / (E_i - E_{i+1}) \qquad (7.13)$$

Cumulative Energy Flux $(E \geq E_{i+1})$

$$\sum_{i'=1}^{i} I_{i'}^o \qquad (7.14)$$

Average Flux Between Outer Iterations n_1 and n_2

$$\frac{1}{n_2 - n_1} \sum_{i=n_1+1}^{n_2} \sum_{k=0} \Delta\phi_{ijk} \qquad (7.15)$$

Flux Contribution from mth Fixed Source

$$\frac{1}{n} \sum_{i=1}^{n} \left\{ \left(\sum_{k=0} \Delta\phi_{ijk} \right) \text{ such that } \vec{r}_{i,o} \text{ is in the m th source} \right\} \qquad (7$$

Flux Contribution from lth Scattering Region

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1} \left\{ \Delta\phi_{ijk} \text{ such that } \vec{r}_{ik} \text{ is in the lth region} \right\} \qquad (7$$

Flux Contribution from kth Order of Scatter

$$\frac{1}{n} \sum_{i=1}^{n} \Delta\phi_{ijk} \qquad ($$

Azimuthally averaged Legendre moments of the angular flux can also be obtain from FASTER. In particular the zeroth moment is the scalar flux given by equation 7.6. first moment is the current. The equations used in obtaining these moments and the equa for externally manipulating these moments are summarized in Appendix B.

Length-of-flight moments of the scalar flux are also computed by FASTER. Thes moments are discussed in Appendix B.

7.3 FLUX CONVERSIONS

Scalar fluxes are converted by group to more useful units using a linear interpo in energy:

$$D_i = \phi_i^o \frac{f_i(\overline{E}_i - E_{i+1}) + f_{i+1}(E_i - \overline{E}_i)}{E_i - E_{i+1}} \qquad ($$

where f_i is the point-wise energy dependent conversion factor or response function for ith energy level.

The total response is also obtained with limits on its relative error:

$$D = \sum_{i=1} D_i \qquad (7.20)$$

$$E_{min} = \frac{1}{D}\left[\sum_i E_i^2 D_i^2\right]^{1/2} \qquad (7.21)$$

$$E_{max} = \frac{1}{D}\sum_i E_i D_i \qquad (7.22)$$

The equation for E_{max} is obtained by applying the Schwarz inequality to the covariance of of the scalar fluxes in individual groups. Total responses are also obtained for each of the flux components and the angular and length-of-flight moments using equations like 7.19 and 7.20, above.

SECTION

8.0 RANDOM SAMPLING TECHNIQUES

The preceding sections detailed the sampling functions in a formal manner only. This section describes the techniques used in the FASTER program for obtaining random discrete position vectors from the sampling functions and the considerations involved in defining these functions.

The development of sampling techniques for the FASTER program has proceeded historically from the approximation of the actual particle distributions to an approximation of optimal sampling functions. Both of these techniques are included in the FASTER program.

8.1 GENERAL SAMPLING PROCEDURES

The technique used by FASTER in selecting discrete values of a random variable x, involves the transformation:

$$p^*(x)\,dx = p(\xi)\,d\xi \qquad (8.1)$$

where ξ is a random variable uniformally distributed on the open interval (0, 1)

$$p(\xi) = \begin{cases} 1, & 0 < \xi < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (8.2)$$

Various numerical techniques are available for randomly selecting discrete values of ξ from this distribution. A function subprogram RANNO is used in FASTER to obtain these discrete values of ξ.

The procedure for randomly selecting discrete values of x then reduces to solving the equation:

$$\int_{-\infty}^{x} p^*(x')\,dx' = \int_0^{\xi} p(\xi')\,d\xi' = \xi \qquad (8.3)$$

Caution is exercised, therefore, to ensure that such a solution can be obtained with relative ease.

The form of $p^*(x)$ frequently used in the FASTER program for one-dimensional *sampling functions involves an exponential function. The probability density function for* x is than written, for x on the internal (a, b), as $p^*(x; a, b, \alpha, x_o)$ with the following special cases:

a) $p^*(x) = \delta(x - a)$ if $a = b$

b) $p^*(x) = \frac{1}{b-a}$ if $a < b$ and $\alpha = 0$

c) $p^*(x) = \frac{\alpha}{c} \exp\left[\alpha \left| x - x_o \right| \right]$ if $a < b$ and $\alpha \neq 0$ (8.4)

where, for $\alpha \neq 0$

x_o is the preferred value of x,

$a \leq x_o \leq b$

$C = A + B$

$$A = \alpha \int_a^{x_o} \exp\left[\alpha \left| x - x_o \right| \right] dx = \exp\left[\alpha \left| a - x_o \right| \right] - 1$$

$$B = \alpha \int_{x_o}^{b} \exp\left[\alpha \left| x - x_o \right| \right] dx = \exp\left[\alpha \left| b - x_o \right| \right] - 1 \qquad (8.5)$$

The parameter α is usually defined from a specification of the relative importance, ρ, of x_o as compared to the point, a or b, farthest away from x_o.

$$\rho = 1/\exp\left[\alpha (x_o - a)\right] \text{ or } \rho = 1/\exp\left[\alpha (b - x_o)\right], \text{ i.e.,}$$

$$\alpha = - \frac{\ln \rho}{\max(x_o - a,\ b - x_o)} \qquad (8.6)$$

The point x_o is "least perferred" if ρ is less than unity.

This probability density function is sampled using the function subprogram SAMPLE in the following manner:

a) <u>if $a = b$</u>

set $x = a$

set $p^*(x) = 1.0$ [actually equal to $\delta(x - a)$] (8.7)

b) <u>if $a < b$ and $\alpha = 0$</u>

obtain ξ from RANNO

solve equation 8.3,

$$\xi = \int_a^x \frac{dx'}{b-a} = \frac{x-a}{b-a}$$

i.e., $x = a + \xi(b - a)$

$p^*(x) = 1/(b - a)$ (8.8)

c) <u>if $a < b$ and $\alpha \neq 0$</u>

obtain ξ from RANNO

solve equation 8.3,

i.e., $\xi = \frac{\alpha}{c} \int_a^x \exp\left[\alpha \left| x - x_o \right| \right] dx$

1) if $\xi < A/C$, then $x < x_o$

and $\xi = \frac{1}{C} \left\{ A - \alpha \int_x^{x_o} \exp\left[\alpha \left| x' - x_o \right| \right] dx' \right\}$

or $\xi C - A = - \left\{ \exp\left[-\alpha (x - x_o)\right] - 1 \right\}$

2) if $\xi > A/C$, then $x > x_o$

and $\xi = \frac{1}{C} \left\{ A + \alpha \int_{x_o}^{x} \exp\left[\alpha \left| x' - x_o \right| \right] dx' \right\}$

or $\xi C - A = \left\{ \exp\left[\alpha (x - x_o)\right] - 1 \right\}$

i.e., let $\delta = \frac{\xi - A/C}{|\xi - A/C|}$ $\begin{cases} +1 \text{ for case 1 above} \\ -1 \text{ for case 2 above} \end{cases}$

$$\left.\begin{array}{ll} \text{then} & \exp\left[\delta\alpha(x - x_o)\right] = 1 + \delta(\xi C - A) \\ \text{let} & D = 1 + \delta(\xi C - A) \\ \text{then} & x' = x_o + \frac{\delta \ln D}{\alpha} \\ \text{and} & p^*(x) = \frac{\alpha D}{C} \end{array}\right\} \quad (8.9)$$

8.2 SPATIAL SAMPLING IN THE FIXED SOURCE COORDINATE SYSTEM

The FASTER program includes both the random selection of initial position vectors in a source centered coordinate system and a sampling procedure utilizing built-in importance functions. However, this latter sampling technique is limited to volume sources. Thus, the first technique is required for problems involving point or line sources. Surface sources can be treated by either technique since they are readily approximated by "thin" volumes.

Random selection of a point in the source coordinate system is performed by subroutine PSTAR. The first step in sampling in the fixed source coordinate system is the random selection of just one of the sources. The following equation defines the total sampling function:

$$p_o^*(\vec{r}) = \sum_{i=1} p_i^* \, q_i^*(\vec{r}) \quad (8.10)$$

where p_i^* is the input relative importance of the ith source

$$\sum_{i=1} p_i^* = 1 \quad (8.11)$$

$$\left.\begin{array}{l} q_i^*(\vec{r}) \quad \text{if } \vec{r} \text{ is in } V_i, \text{ the ith source volume} \\ q_i^*(\vec{r}) = 0 \text{ if } \vec{r} \text{ is not in } V_i \\ \iiint q_i^*(\vec{r})\, dV = 1 \end{array}\right\} \quad i = 1, 2, \ldots \quad (8.12)$$

The appropriate volume is then selected using a random number ξ from RANNO

$$\xi = \iiint p_o^*(\vec{r})\, dV = \sum_{i'=1}^{i-1} p_{i'}^* + p_i^* \iiint_{fV_i} q_i^*(\vec{r})\, dV \quad (8.13)$$

where f is some fraction, less than 1.0, of the volume V_i. Thus, the point $\vec{r}$ is in the fixed source volume i for which

$$\sum_{i'=1}^{i-1} p_{i'}^* < \xi \le \sum_{i'=1}^{i} p_{i'}^* \quad (8.14)$$

Having selected the fixed source, then each of the spatial variables v_1, v_2, v_3 is obtained using the sampling function SAMPLE, described in Section 8.1

$$q_i^*(\vec{r})\, dV = \prod_{j=1}^{3} p^*\left(v_j;\, v_j^{min},\, v_j^{max},\, v_j^o,\, \alpha_j\right) \frac{dV}{v_1^{\,n}} \quad (8.15)$$

where
- n = 1 and 2 for cylindrical and spherical geometries, respectively
- = 0 for rectangular geometries
- v_j^{min} is the minimum value of the jth variable
- v_j^{max} is the maximum value of the jth variable
- v_j^o is the preferred value of the jth variable
- $\alpha_j = \dfrac{-\ln \rho_j}{\max\left(v_j^o - v_j^{min},\, v_j^{max} - v_j^o\right)}$
- ρ_j is the input relative importance of v_j^o

The $1/v_1^n$ factor is introduced to simplify the difference in differential volume elements for the different source geometries:

$$\frac{dV}{v_1^n} = \left(v_1^n \, dv_1 \, dv_2 \, dv_3\right)/v_1^n = dv_1 \, dv_2 \, dv_3 \qquad (8.16)$$

The rectangular components of $\vec{r}_o$ are then obtained by the variable transformations discussed in Section 5.0. Since the point $\vec{r}_o$ defined by (v_1, v_2, v_3) is in the ith volume, the value of $p_o^*(\vec{r}_o)$ is:

$$p_o^*(\vec{r}_o) = p_i^* \, q_i^*(\vec{r}_o) \qquad (8.17)$$

8.3 SOURCE SAMPLING USING A PSEUDO SPHERICAL SOURCE

Another sampling procedure was developed for fixed volume sources and attempts to minimize the variance associated with the source point selection. The function approximates equation 2.36 by a sampling function with a separable angular and spatial dependence.

$$p_o^*(\vec{r}) \, dV = \frac{u^*(\vec{\Omega}') \, v^*(s;\vec{\Omega})}{s^2} \left(s^2 \, ds \, d\Omega'\right) \qquad (8.18)$$

where a transformation to a spherical coordinate system about a preferred point $\vec{r}_p$, e.g., a point detector, has been performed. The variables involved in this sampling function are shown in Figure 5. The random selection procedures are incorporated in subroutine SPHERE.

Angular Dependence

The z-axis of the spherical coordinate system is directed towards the center of the sources, i.e., towards $\vec{r}_c$ a point in the center of a sphere, of radius R, which encloses all the fixed sources. The random selection of the direction vector $\vec{\Omega}'$ in this rotated coordinate system uses the sampling function:

$$u^*(\vec{\Omega}')d\Omega' = p^*\left(\mu'; \mu^{min}, 1, 1, \alpha_{\mu'}\right) p^*\left(\theta'; -\pi, \pi, 0, \alpha_{\theta'}\right) d\mu' \, d\theta' \qquad (8.19)$$

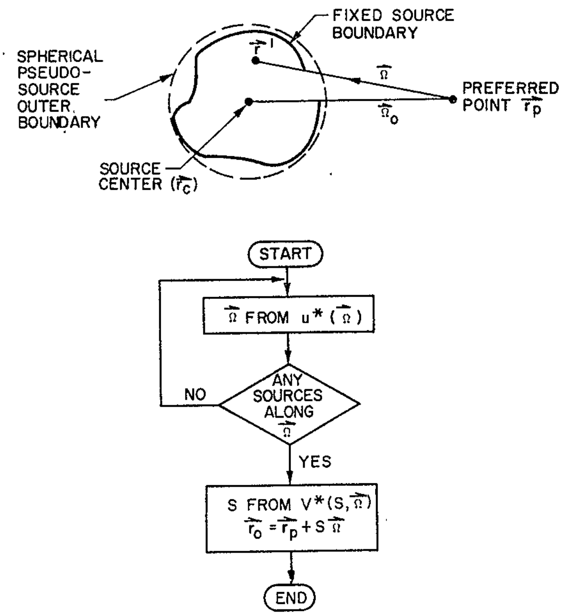


Figure 5. Optimal Fixed Source Sampling

i.e., discrete values of μ' and are obtained from the function subprogram SAMPLE. The *azimuthal angle varies over all possible values. The minimum value of the cosine of the* polar angle is defined as:

$$\mu^{min} = -1 \text{ if } R > R_o = \left|\vec{r}_p - \vec{r}_c\right|$$

$$= \left[R_o^2 - R^2\right]^{1/2} / R_o \text{ if } R \leq R_o \qquad (8.20)$$

i.e., the angular variables are limited to directions which intercept the sphere enclosing the fixed sources. The parameters $a_{\mu'}$ and $a_{\theta'}$ are defined from the input relative importances: $\rho_{\mu'}$ and $\rho_{\theta'}$:

$$a_{\mu'} = \frac{-\ln \rho_{\mu'}}{1 - \mu^{min}}, \quad \rho_{\mu'} = \frac{\text{importance of } \mu' = 1}{\text{importance of } = \mu'^{\,min}}$$

$$a_{\theta'} = \frac{-\ln \rho_{\theta'}}{\pi}, \quad \rho_{\theta'} = \frac{\text{importance of } \theta' = 0}{\text{importance of } \theta' = \pm\pi} \qquad (8.21)$$

The values of μ' and θ' obtained through this sampling procedure define the components of a direction vector in a rotated coordinate system

$$\vec{\Omega}' = c_1' \vec{i}' + c_2' \vec{j}' + c_3' \vec{k}' \qquad \begin{aligned} c_1' &= \sqrt{1-\mu'^2}\cos\theta' \\ c_2' &= \sqrt{1-\mu'^2}\sin\theta' \\ c_3' &= \mu' \end{aligned} \qquad (8.22)$$

These components are then transformed into equivalent values in the geometry coordinate system:

$$\vec{\Omega} = c_1 \vec{i} + c_2 \vec{j} + c_3 \vec{k} \quad \text{where } c_i = \sum_{j=1}^{3} c_j' R_{j,i} \quad i = 1, 2, 3 \qquad (8.23)$$

The rotation matrix $R_{j,i}$ is obtained as shown in Figure 4 for the rotation angles:

$$\phi_r = \cos^{-1}(c_3^o)$$

$$\theta_r = \tan^{-1}(c_2^o/c_1^o) \qquad (8.24)$$

where $\vec{\Omega}_o = c_1^o \vec{i} + c_2^o \vec{j} + c_3^o \vec{k} = (\vec{r}_c - \vec{r}_p)/k$, the unit vector directed towards the center of the sphere enclosing the fixed sources.

<u>*Spatial Dependence*</u>

The spatial sampling function incorporated in subroutine SPHERE is defined for all direction vectors $\vec{\Omega}$ even if no sources lie along $\vec{r}_p + s\vec{\Omega}$ for $s \geq o$. If no source is intercepted along this ray, the outer iteration will yield a zero result. To ensure an adequate number of outer iterations, discrete directions are obtained by ramdom sampling until one of the defined rays intercepts a fixed source. The misses are counted internally and factored into the flux averages; thus if n' is the requested number of iterations, $n \geq n'$ iterations will be performed until n' non-zero contributions are obtained.

The spatial sampling function is defined as: for s in the <u>ith</u> region along Ω as:

$$v^*(s,\vec{\Omega}) = \frac{A_i \exp\left[-a_i(s - s_i)\right]}{\sum_i \frac{A_i}{a_i}\left[1 - \exp\left[-a_i(s_{i+1} - s_i)\right]\right]} \qquad (8.25)$$

where $A_i = 0$, if there is no source in the region

$$A_i = I_{i'} P_{i'}\left(\vec{r}_p + (s_i + \epsilon)\vec{\Omega}\right) \exp\left[-\int_o^{s_i} \Sigma_{\hat{t}_o}^{t}(\vec{r}_p + s'\vec{\Omega})\, ds'\right] \qquad (8.26)$$

if there is a source, i', in the region

$$a_i = \beta_s \Sigma_{\hat{t}_o}^{t}(\vec{r}_p + \bar{s}_i\vec{\Omega}) + \frac{2}{s_{i+1} - s_i} \ln\left[\frac{P_{i'}\left[\vec{r}_p + (s_i + \epsilon)\vec{\Omega}\right]}{P_{i'}\left(\vec{r}_p + \bar{s}_i\vec{\Omega}\right)}\right] \qquad (8.27)$$

$$\bar{s}_i = (s_i + s_{i+1})/2$$

$p_{i'}$ $(\vec{r}_p + (s_i + \epsilon)\vec{\Omega})$ is the normalized spatial source density just inside the region,

i.e. $\epsilon = \frac{s_{i+1} - s_i}{100}$

$p_{i'}$ $(\vec{r}_p + \bar{s}_i \vec{\Omega})$ is the normalized spatial source density half way through the region

$I_{i'}$ is the total source intensity (Mev/sec),

$\hat{i}_o$ is an average energy group for the sources (described below),

β_s is an input adjustment factor for the material attenuation importance, and a two point exponential curve fit of the spatial source density with distance has been used

The average source group index, $\hat{i}_o$, corresponds to an average source energy $\bar{E}_p$ computed for the preferred point $\vec{r}_p$ by subroutine GROUP as:

$$E_{\hat{i}_o} > \bar{E}_p \geq E_{\hat{i}_o + 1}$$

$$\text{where } \bar{E}_p = \frac{\sum_i G_i \overline{E_i^o}}{\sum_i G_i} \quad , \qquad G_i = \frac{g_i \, n_i^o \exp[-\eta_i] \; (1 + b_i \eta_i)}{\Sigma_i^v} \qquad (8.28)$$

where g_i is an input group importance e.g., a flux-to-dose conversion factor

b_i is an input linear buildup coefficient,

n_i^o, $\overline{E_i^o}$ define the spectrum of the first source encountered on the line from $\vec{r}_c$ to $\vec{r}_p$ (the source closest to $\vec{r}_c$),

η_i is the number of mean free paths from the edge of this source nearest to the preferred point $\vec{r}_p$,

Σ_i^v is the cross section for the region over which the source is superimposed ($\equiv 1.0$ if the region is void),

The spatial function is sampled by obtaining a random number ξ from RANNO and then solving the equivalent of equation 8.3.

$$\xi = \int_o^s v^*(s'; \vec{\Omega})\, ds'$$

$$= \frac{\sum_{i=1}^{k-1} \frac{A_i}{\alpha_i} \left[1 - \exp\left[-\alpha_i (s_{i+1} - s_i)\right]\right] + \int_{s_k}^{s} A_k \exp\left[-\alpha_k (s' - s_k)\right] ds'}{\sum_i \frac{A_i}{\alpha_i} \left[1 - \exp\left[-\alpha_i (s_{i+1} - s_i)\right]\right]} \qquad (8.29)$$

$$\left.\begin{aligned} s &= s_k - \frac{1}{\alpha_k} \ln \left[1 - \frac{\alpha_k}{A_k} (\xi A_{>o} - A_{<k})\right] \\ v^*(s; \vec{\Omega}) &= \left[A_k - \alpha_k \left(\xi A_{>o} - A_{<k}\right)\right] / A_{>o} \end{aligned}\right\} \qquad (8.30)$$

where $A_{>o} = \sum_i \frac{A_i}{\alpha_i} \left[1 - \exp\left[-\alpha_i (s_{i+1} - s_i)\right]\right]$, the denominator of equation 8.29

$A_{<k} = \sum_{i=1}^{k-1} \frac{A_i}{\alpha_i} \left[1 - \exp\left[-\alpha_i (s_{i+1} - s_i)\right]\right]$, that part of the numerator of equation 8.29 up to the region containing the source point

Thus the discrete position reactor is given by·

$$\vec{r}_o = \vec{r}_p + s\vec{\Omega}$$

$$\text{with } p_o^*(\vec{r}_o) = \frac{u^*(\vec{\Omega}')\, v^*(s; \vec{\Omega})}{s^2} \qquad (8.31)$$

8.4 SEPARATION OF VARIABLES FOR SCATTERING POINT SELECTIONS

Application of the procedures discussed in Sections 8.3 and 8.4 reduces the representation of the fixed source(s) to a single point source. The random sampling techniques for selecting the kth scattering point $\vec{r}_k$ therefore involve a point angular source at $\vec{r}_{k-1}$ and a preferred point at $\vec{r}_p$ as shown in Figure 6. For point flux calculations in source and/or scattering volumes, the preferred point $\vec{r}_p$ is the detector point $\vec{r}$.

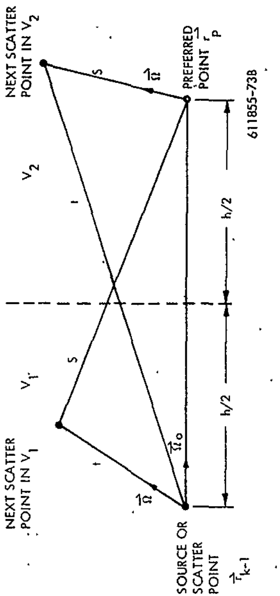


Figure 6. Optimal Scattered Source Sampling

Surface and volume averaged flux calculations utilize the preferred point $\vec r_p$ to locate in a general manner the area in space where the detectors are located. For these calculations the point $\vec r_p$ is assumed to be at the center of a sphere of radius R_p which contains all the detectors. It is introduced to permit a more comprehensive treatment of the spatial importance of scattering points.

The first technique used in selecting the scattering point is to reduce equation 2.37 to a more simple form by using a representative energy group and by defining the sampling function as a separable function of direction and distance:

$$p_k^*(\vec r')\,dV = \frac{q^*(\vec\Omega')\,\upsilon^*(x)\,dV}{x^2} \qquad (8.32)$$

where both $q^*(\vec\Omega')$ and $\upsilon^*(x)$ depend on $\hat i$, the index of the energy group corresponding to an average group energy determined from the flux estimation performed for the source point $\vec r_{k-1}$:

$$E_{\hat i} > \bar E \geq E_{\hat i+1}$$

$$\bar E = \frac{\sum_{detectors}\sum_{j} \Delta\phi_{i,j,k-1}\, g_j\, \eta_j\, b_j\, \bar E_j^s}{\sum_{detectors}\sum_{j} \Delta\phi_{i,j,k-1}\, g_j\, \eta_j\, b_j} \qquad (8.33)$$

The separation of the sampling function into two components permits the selection of *a discrete direction vector* $\vec\Omega'$ *with a cursory treatment* of geometric effects. The distance is then selected from the second component using a more detailed treatment of the geometric effects.

It should be noted that the angular integration of the point sources used in obtaining surface and volume averaged fluxes involves a repetitive sampling of the angular part of this sampling function; i.e., discrete directions $\vec\Omega_l$ are randomly selected from the angular sampling function $q^*(\vec\Omega')$.

For surface and volume flux calculations the discrete direction and distance defining $\vec{r}_k$ are obtained in a spherical coordinate system which is always centered at the source point $\vec{r}_{k-1}$ i.e. $dV = t^2 dt d\Omega$. A spherical coordinate system is also used for individual point detectors, except the origin can be located at the detector point $\vec{r}_p$ with $dV = s^2 ds d\Omega'$.

The considerations involved in selecting the origin for point detectors incorporate the fact that there are two singularities in the sampling function, $1/t^2$ from the source point and $1/s^2$ from the detector. Both singularities must be accounted for while also treating material distributions, etc. The singularity at the selected coordinate system origin is removed by the differential volume element. A very simple technique for the second singularity involves the definition:

$$u^*(\vec{r}\,';\Omega)\,dV = \left\{P_1^*\; u_1^*(\vec{r}\,') + P_2^*\; u_2^*(\vec{r}\,')\right\} dV \qquad (8.34)$$

where

$u_1^*(\vec{r}\,') = 0$ for points in V_2

$u_2^*(\vec{r}\,') = 0$ for points in V_1

V_1 is the volume on the source side of a plane perpendicular to and bisecting the line between the source and detector, and V_2 is the volume on the other side of this plane.

The appropriate volume, and coordinate system origin, is randomly selected using the criterion that the relative importance of scattering in each of the volumes is proportional to the mean free paths encountered in the volumes along the line between the source and detector points:

$P_1^* = \eta_1/\eta$, the assumed importance of volume V_1

$P_2^* = \eta_2/\eta$, the assumed importance of volume V_2

$\eta = \eta_1 + \eta_2$

where $\eta_1 = \int_0^{h/2} \Sigma_{\hat{t}}^{t}(\vec{r}_{k-1} + t'\vec{\Omega}_k)\,dt'$, an average number of mean free paths in volume V_1

$$\eta_2 = \int_0^{h/2} \Sigma_{\hat{t}}^{t}(\vec{r}_p - s'\vec{\Omega}_k)\,ds' \text{ , an average number of mean free paths in volume } V_2 \qquad (8.35)$$

The appropriate volume is selected using a random number ξ on (0, 1).

V_1 if $\xi = \iiint P_k^*(\vec{r}\,')\,dV = f \cdot P_1^*$ i.e. $\xi < P_1^*$

$$V_2 \text{ if } \xi = P_1^* + f' P_2^* \text{ i.e. } \xi > P_1^* \qquad (8.36)$$

where f and f' are fractions. If P_1^* is zero, the point $\vec{r}_p$ is automatically selected as the origin and the definition of the $u_2^*(\vec{r}\,')$ is extended over all space. Similarly if P_2^* is zero, the point $\vec{r}_{k-1}$ is selected as an origin with the definition of $u_1^*(\vec{r}\,')$ being extended to all space points.

This procedure does not eliminate the consideration of the second singularity. It does, however, restrict it to values greater than $4/|\vec{r}_{k-1} - \vec{r}_p|^2$, which is sufficient to remove the major difficulty.

8.5 DIRECTION TO SCATTERING POINT

Direction Vector Sampling-Source Coordinate System

There is a special form of $q^*(\vec{\Omega}')$ available for the case of $k = 1$; i.e. when $\vec{r}_{k-1}$ is the original point in the fixed source. Since the capability of specifying angular sources is still permitted, these directions may require random sampling in the source coordinate system. In particular, a monodirectional source will always require this procedure.

The spatial part of the sampling function is then defined for all space, the source point is the origin, and the direction vector in the rotated source coordinate system is obtained by subroutine QSTAR using SAMPLE.

$$q^*(\vec{\Omega}')\,d\Omega' = \prod_{i=4}^{5} p^*(v_i; v_i^{min}, v_i^{max}, v_i^{o}, a_i)\,dv_i \qquad (8.37)$$

where v_4 is the azimuthal angle, v_5 is the cosine of the polar angle, and the other quantities have the same meaning as in equation 8.15. This is analogous to the procedure used for spatial sampling in the source coordinate system as discussed in Section 8.2. The selected variables v_4 and v_5 define the direction vector through a rotation procedure similar to that used in previous discussions.

<u>Direction Vector Selection - General</u>

The general procedure for selecting discrete directions attempts to include the importance of the scattered direction, as measured from the direction before scattering, and as measured from the unit vector towards the preferred point.

The unnormalized angular dependence for the scattering angle incorporates an assumed exponential variation with the cosine of the scattering angle:

$$f_a\left(\vec{\Omega} \cdot \vec{\Omega}_{k-1}\right) = f_a(\mu) = \exp\ [\alpha_s \bar{\alpha}_s \mu]$$

where

$$\left.\begin{aligned} \bar{\alpha}_s &= \left(\rho^h \sigma_{\hat{i}}^h \alpha_{\hat{i}}^h - \Sigma_{\hat{i}}^t \alpha_{\hat{i}}^H\right) \Big/ \left(\rho^h \sigma_{\hat{i}}^h + \Sigma_{\hat{i}}^t\right) \text{ in subroutine SOBER} \\ \bar{\alpha}_s &= \left(\eta^h \alpha_{\hat{i}}^h + (\eta - \eta^h)\,\alpha_{\hat{i}}^H\right) \Big/ \eta \quad \text{in subroutine SOLVER} \end{aligned}\right\} \qquad (8.38)$$

η^h is the number of mean free paths in hydrogen between $\vec{r}_{k-1}$ and $\vec{r}_p$

ρ^h is the local hydrogen density

$\sigma_{\hat{i}}^h$ is the total microscopic hydrogen cross section for group $\hat{i}$

$\Sigma_{\hat{i}}^t$ is the total cross section—except hydrogen—for group $\hat{i}$

$\alpha_i^h = -\frac{1}{2} \ln \rho_i^h$

ρ_i^h is an input relative importance of forward-to-backward scattering for the general group for hydrogen

$\alpha_i^H = -\frac{1}{2} \ln \rho_i^H$

ρ_i^H is the forward-to-backward scatter importance for heavy elements

α_s is an input parameter used to increase or decrease this angular effect.

The unnormalized angular dependence with respect to the preferred point—the effect of the material attenuation—is also assumed to vary exponentially.

$$f_p(\vec{\Omega} \cdot \vec{\Omega}_p) = \exp(\alpha_p \bar{\alpha}_p \vec{\Omega} \cdot \vec{\Omega}_p) \qquad (8.39)$$

where

α_p = is an input adjustment factor,

$\bar{\alpha}_p$ = is an internally computed number

The latter number $\bar{\alpha}_p$ is obtained from a one velocity approximation of the scattering point importance which, by neglecting unimportant constants has the forms:

$$f_p(\mu) = \int_0^{s_{max}} \frac{\exp\left[-\bar{\Sigma}(s+t)\right]}{s^2 t^2}\, s^2 ds \qquad (8.40)$$

where

$$\Sigma = \eta/h \quad \eta = \eta_1 + \eta_2$$

$$s_{max} = h/2\mu, \; \mu > 0$$

$$= \infty, \; \mu \leq 0$$

$$h = |\vec{r}_{k-1} - \vec{r}_p|$$

Thus $$f_p(\mu)\Big|_{\mu=1} = \int_0^{h/2} \frac{\exp[-\Sigma h]}{(h-s)^2} ds = \frac{e^{-\Sigma h}}{h} = \frac{e^{-\eta}}{h}.$$

and $$f_p(\mu)\Big|_{\mu=-1} = \int_0^{\infty} \frac{\exp[-\Sigma h - 2\Sigma s]}{(h+s)^2} ds \leq \frac{e^{-\eta}}{h} [12\eta]^{-1/2}$$

where Schwarz' inequality was used.

Letting $f_p(1)/f_p(-1) = \exp 2\bar{a}_p$

Then $$\exp[2\bar{a}_p] \geq \sqrt{12\eta} \quad \text{or} \quad \bar{a}_p \geq \frac{1}{4} \ln(12\eta) \tag{8.41}$$

The equality is used in the FASTER program.

The effect of this parameter has been examined for $\mu = 1$ by computing:

$$\frac{\partial f(\mu)}{\partial \mu}\Big|_{\mu=1} = \left[\frac{\partial}{\partial \mu} \int_0^{h/2\mu} \frac{\exp[-\Sigma(s+t)]}{t^2} ds\right]_{\mu=1} = \frac{e^{-\eta}}{h}\left(\frac{\eta}{2} - \frac{1}{3}\right)$$

where Leibnitz' rule was applied, μ was then set equal to 1, and then the integration performed. If the true behavior at $\mu = 1$ was exponential, then

$$\bar{a}_p = \frac{\eta}{2} - \frac{1}{3} \tag{8.42}$$

It is noted that for large values of η, this is a stronger angular dependence than that given by equation 8.41.

The total angular dependence of the sampling function is approximated by various combinations of the above functions. The angular variables are obtained using the function subprogram SAMPLE.

a) Towards Detector Most Important

$$q^*(\vec{\Omega}') = p^*(\mu'; -1, 1, -1, a_p \bar{a}_p)$$
$$p^*\left(\theta' - \theta_o; -\pi, \pi, 0, -a_s \bar{a}_s \sqrt{1-\mu'^2} \sqrt{1 - \vec{\Omega}_{k-1} \cdot \vec{\Omega}_p^{\,2}}\right) \tag{8.43}$$

where

$$\vec{\Omega}_r = \vec{\Omega}_p$$

$$\mu' = \vec{\Omega}' \cdot \vec{\Omega}_r$$

and θ_o is the azimuthal angle between the $\vec{i}'$ axis and the projection of $\vec{\Omega}_{k-1}$ in the $x' - y'$ plane and where the rotated coordinate system has $\vec{k}' = \vec{\Omega}_p$.

b) Original Direction Most Important

$$q^*(\vec{\Omega}') = p^*(\mu'; -1, 1, -1, a_s \bar{a}_s)$$
$$p^*\left(\theta' - \theta_o; -\pi, \pi, 0, -a_p \bar{a}_p \sqrt{1-\mu'^2} \sqrt{1 - \vec{\Omega}_{k-1} \cdot \vec{\Omega}_p^{\,2}}\right) \tag{8.44}$$

where

$$\vec{\Omega}_r = \vec{\Omega}_{k-1}$$

$$\mu' = \vec{\Omega}' \cdot \vec{\Omega}_r$$

θ_o is the azimuthal angle between the $\vec{i}'$ axis and the projection of $\vec{\Omega}_p$ in the $x'-y'$ plane, and where the rotated coordinate system has $\vec{k}' = \vec{\Omega}_{k-1}$.

Only the latter technique, equation 8.44, is generally applicable to individual point detectors since the cosine of the polar angle μ' used in the first technique, equation 8.43, applies only if the source point $\vec{r}_{k-1}$ is selected as the coordinate system origin. As a matter of fact, both of these techniques neglect the effect of the scattering angle at $\vec{r}'$, the next scattering point being selected.

The effects of the scattering angle $\cos^{-1}\mu''$ at the next scattering point involves the use of the unnormalized approximation:

$$f(\mu'')\Big|_{\mu=1} = \int_0^{h/2\mu} \frac{\exp\left[a_s\bar{a}_s\mu''\right]}{t^2}\,ds\Big|_{\mu=1} = \frac{\exp\left[a_s\bar{a}_s\right]}{h}.$$

$$f(\mu'')\Big|_{\mu=-1} = \int_0 \frac{\exp\left[a_s\bar{a}_s\mu''\right]}{t^2}\,ds\Big|_{\mu=-1} = \frac{\exp\left[-a_s\bar{a}_s\right]}{h}$$

Assuming an exponential variation $\exp\left[\bar{a}\mu\right]$ for intermediate angles μ implies

$$\bar{a} = \bar{a}_s \qquad (8.45)$$

This parameter can be scaled through the input number a_s.

Combining this variation with that for material attenuation effects, equation 8.41 or 8.42 yields a third technique for selecting the direction vector with more or less equal applicability to either selected origin in point flux calculations:

c) Combination

$$q_i^*(\vec{\Omega}') = p^*(\mu'; -1, 1, -1, a_s\bar{a}_s - a_p\bar{a}_p)\; p^*(\theta', -\pi, \pi, 0, 0) \qquad (8.46)$$

where $\vec{\Omega}_r = \vec{\Omega}_p$

$\mu' = \vec{\Omega}' \cdot \vec{\Omega}_p$

θ' is a uniformly distributed azimuthal angle

All of the above angular sampling techniques, equations 8.43, 8.44, and 8.46, yield polar angle cosines, μ', and azimuthal angles θ' in a rotated coordinate system. They require application of a rotation to be reduced to equivalent values in the geometry coordinate system. The z' - axis of the rotated coordinate system is $\vec{\Omega}_r$ as indicated above. The rotations are performed by previously described technqiues.

8.6 DISTANCE-TO-SCATTERING POINT SELECTION

The spatial sampling function can contain a more detailed treatment of the effects of material distributions since it is defined for a fixed direction. For this discussion, the following notation will be adopted:

$\vec{r}_a$ is the position vector of the selected origin, either $\vec{r}_{k-1}$ or $\vec{r}_p$

$\vec{r}_b$ is the position vector of the other discrete point

h is the distance from $\vec{r}_a$ to $\vec{r}_b = \left|\vec{r}_b - \vec{r}_a\right|$

$\vec{\Omega}$ is the unit direction vector from $\vec{r}_a$ to $\vec{r}_b$ $= (\vec{r}_b - \vec{r}_a)/h$

$\vec{\Omega}'$ is the direction vector selected from the angular sampling function.

μ_o is the cosine of the angle between $\vec{\Omega}$ and $\vec{\Omega}'$ $= \vec{\Omega} \cdot \vec{\Omega}'$

s_t is the distance from $\vec{r}_a$ along $\vec{\Omega}'$ to the outer boundary of the geometry,

s_m is the maximum distance along $\vec{\Omega}'$ which can be considered for a scattering point.

$s_m = \min(s_t, h/2\mu_o)$ if sampling is restricted to the half space around $\vec{r}_a$ and if $\mu_o > 0$

$s_m = s_t$ in all other situations.

$\Sigma^{v}_{\hat{i}}(s)$ is the total cross section at a distance s from $\vec{r}_a$ for the energy group $\hat{i}$, *also denoted by* $\Sigma^{v}_{\hat{i},i'}$ *where* i' *indicates the composite* material located at this distance.

α is an adjustment factor for material attenuation importance along the first leg of the scattering triangle (from $\vec{r}_a$).

β is a similar adjustment factor for the second leg of this triangle (from $\vec{r}_b$).

<u>Exponential Transformation</u>

There are two techniques available in subroutine USTAR for selecting the distance to the scattering point. The first is the "exponential transformation". There is no essential difficulty with using this technique for point detectors since the singularity is removed by the procedures used in selecting the half space.

The exponential transformation is written in the form:

$$u^{*}(s) = \frac{-\frac{d}{ds}\exp\left[-\int_{o}^{s}\Sigma^{v}_{\hat{i}}(s')\,[\alpha-\beta\mu(s')]ds'\right]}{1-\exp\left[-\int_{o}^{s_m}\Sigma^{v}_{\hat{i}}(s')\,[\alpha-\beta\mu(s')]ds'\right]} \qquad (8.47)$$

where

$\mu(s)$ is the cosine of the scattering angle to $\vec{r}_b$ at a distance s from $\vec{r}_a$, i.e., at $\vec{r}' = \vec{r}_a + s\vec{\Omega}'$

$$\mu(s) = \frac{\mu_o h - s}{t}$$

t is the length of the second leg of the scattering triangle

$$= \sqrt{h^2 + s^2 - 2\mu_o sh}$$

The extent to which this sampling function approximates equation 2.37 for the fixed direction $\vec{\Omega}'$ can be determined by performing the differentiation

$$u^{*}(s) = \frac{1}{C}\,\Sigma^{v}_{\hat{i}}(s)[\alpha-\beta\mu(s)]\exp\left[-\int_{o}^{s}\Sigma^{v}_{\hat{i}}(s')\,[\alpha-\beta\mu(s')]ds'\right] \qquad (8.48)$$

where C is the denominator of equation 8.47. The coefficient in front of the exponential *should be approximating a term* $t^{-2}\,d\Sigma\,[s,\mu(s)]/d\Omega$ *except for a normalizing constant.* Its actual form is:

$$\Sigma^{v}_{\hat{i}}(s)\,[\alpha-\beta\mu(s)] = \Sigma^{v}_{\hat{i}}(s)\left(\alpha-\beta\,\frac{\mu_o h-s}{t}\right)$$

$$= \Sigma^{v}_{\hat{i}}(s)\,\frac{\left[(\alpha t+\beta s)-\beta\mu_o h\right]}{t} \qquad (8.49)$$

which is not very close to the desired representation. It does include an approximation of the scattering cross section by the total cross section, however.

The exponential term can be examined best by performing the integration of the argument:

$$\int_{o}^{s}\Sigma^{v}_{\hat{i}}(s)\,(\alpha-\beta\mu(s))\,ds = \alpha\sum_{i}\Delta s_i\,\Sigma^{v}_{\hat{i},i'} - \beta\sum_{i}\Sigma^{v}_{\hat{i},i'}\int_{s_i}^{s_{i+1}}\mu(s)\,ds$$

$$= \alpha\sum_{i}\Delta s_i\,\Sigma^{v}_{\hat{i},i} + \beta\sum_{i}\Delta t_i\,\Sigma^{v}_{\hat{i},i} \qquad (8.50)$$

where

Δs_i is the incremental distance in the <u>ith</u> region along $\vec{\Omega}$

$= s_{i+1} - s_i$

Δt_i is the change in distance on the other leg of the scattering triangle corresponding to Δs_i, i.e.,

$$\Delta t_i = -\int_{s_i}^{s_{i+1}} \mu(s)\,ds = t(s)\Big|_{s_i}^{s_{i+1}} = t_i - t_{i+1}$$

The first term in this expression properly accounts for attenuation along the first leg of the scattering triangle. The second term accounts for the relative effect of the change in attenuation along the second leg with an implicit assumption of material distributions being spherically symmetric about $\vec{r}_b$. This assumption is as good as any since a detailed treatment of the second leg of the triangle is prohibitive.

Thus, this sampling function approximates to some degree the desired effects. It is sampled by obtaining a random number ξ from RANNO and solving the equivalent of equation 8.3:

$$\xi = \frac{1}{C}\left[1 - \exp\left(-\int_0^s \Sigma_{\hat{i}}^{v}(s')\,(\alpha - \beta\mu(s'))\,ds'\right)\right]$$

i.e. $$\exp\left[-\int_0^s \Sigma_{\hat{i}}^{v}(s')\left(\alpha - \beta\mu(s')\right)ds'\right] = 1 - \xi C$$

or $$\left[\alpha(s - s_i) + \beta(t - t_i)\right]\Sigma_{\hat{i},i'}^{v} + \sum_{i=1}^{i-1}\Sigma_{\hat{i},i''}^{v}\,(\alpha\Delta s_i + \beta\Delta t_i) = -\ln(1 - \xi C)$$

This equation can be written in the form

$$t = As + B$$

where

$$A = -\frac{\alpha}{\beta}$$

$$B = -\frac{1}{\beta}\left\{\alpha s_i + \beta t_i - \frac{1}{\Sigma_{\hat{i},i'}^{v}}\left[\ln(1-\xi C) + \sum_{i=1}^{i-1}\Sigma_{\hat{i},i''}^{v}(\alpha\Delta s_i + \beta\Delta t_i)\right]\right\}$$

But $$t^2 = s^2 + h^2 - 2\mu_o sh = A^2 s + 2ABs + B^2$$

or $$A's^2 + 2B's + C' = 0 \qquad A' = A^2 - 1$$

$$B' = AB + \mu_o h$$

$$C' = B^2 - h^2$$

Thus

$$s = \frac{-B' + \sqrt{B'^2 - A'C'}}{A'} \tag{8.51}$$

$$p^*(s) = \Sigma_{\hat{i}}^{v}(s)\left(\alpha - \beta\,\frac{\mu_o h - s}{As + B}\right)\left(1 - \xi C\right)$$

Curve Fit of Spatial Dependence ($1/t^2$)

The second technique used for selecting the distance to the scattering point is similar in some respects to the above function. In an unnormalized form, it attempts to approximate equation 2.37 by

$$u^*(s) = \Sigma_{\hat{i}}^{v}(s)\,\Delta\Omega(s)\exp\left[-\int_0^s \Sigma_{\hat{i}}^{v}(s')[\alpha - \beta\mu(s')]\,ds'\right] \tag{8.52}$$

where $\Delta\Omega(s)$ is the $1/t^2$ factor for point detector calculations, and is a fractional solid angle for surface and volume detector calculations.

This function is curve-fit from boundary to boundary of the regions traversed by:

$$u^*(s) = A_i \exp\left[-\alpha_i(s - s_i)\right] \quad \text{for } s_i < s < s_{i+1} \tag{8.53}$$

where $$u^*(s_i) = \Sigma_{\hat{i},i'}^{v}\,\Delta\Omega(s_i)\exp\left[-\int_0^{s_i}\Sigma_{\hat{i}}^{v}(s')[\alpha - \beta\mu(s')]ds'\right]$$

$$u^*(s_{i+1}) = \Sigma_{\hat{i},i'}^{v}\,\Delta\Omega(s_{i+1})\exp\left[-\int_0^{s_{i+1}}\Sigma_{\hat{i}}^{v}(s')[\alpha - \beta\mu(s')]ds'\right]$$

i.e.,

$$A_i = u^*(s_i)$$

$$\alpha_i = -\frac{1}{s_{i+1}-s_i} \quad \ln \left[u^*(s_{i+1}) / u^*(s_i)\right]$$

$$= \alpha\, \Sigma^{v}_{\hat{i},i'} - \frac{1}{s_{i+1}-s_i} \left[\ln \frac{\Delta\Omega(s_{i+1})}{\Delta\Omega(s_i)} - \beta\, \Sigma^{v}_{\hat{i},i}(t_{i+1}-t_i)\right] \qquad (8.54)$$

The fractional solid angle is computed for the detector sphere in surface and volume flux calculations as:

$$\Delta\,\Omega(s_i) = 1 \text{ if } t_i < R_p$$

$$= \frac{1}{4} \left(\frac{R_p}{t_i}\right)^2 \left[1 + \frac{1}{4}\left(\frac{R_p}{t_i}\right)^2\right] \quad \text{if } t_i > R_p \qquad (8.55)$$

This is a second order Taylor series expansion and is sufficiently accurate considering the other approximations.

This function is sampled by the transformation equivalent to equation 8.3 using a random number ξ from RANNO:

$$\xi = \frac{\int_0^s u^*(s')ds'}{\int_0^{s_m} u^*(s')ds'} = \frac{1}{C} \left\{\sum_{i'=1}^{i-1} \frac{A_{i'}}{\alpha_{i'}} \left[1 - \exp\left[-\alpha_{i'}\,\Delta s_{i'}\right]\right] + \int_{s_i}^{s} A_i \exp\left[-\alpha_i\,(s'-s_i)\right] ds'\right\}$$

or

$$A_i \exp\left[-\alpha_i(s-s_i)\right] = A_i - \alpha_i \left\{\xi C - \sum_{i'=i}^{i-1} \frac{A_{i'}}{\alpha_{i'}} \left[1 - \exp(-\alpha_{i'}\,\Delta s_{i'})\right]\right\} = D$$

where D is the right side of the above equation and C is the denominator of the original.

Thus $u^*(s) = D/C$

and $s = s_i - \frac{1}{\alpha_i} \ln(D/A_i)$ (8.56)

SECTION

9.0 DATA INPUT INSTRUCTIONS

9.1 GENERAL INPUT PROCEDURES

Input Formats

The FASTER program utilizes standard Fortran input statements, Reference 16 page 19. A variety of formats are used. Each format utilizes various combinations of the following data fields

hollerith information: A4 field (4 columns)

integer data: I3 field (3 columns)

floating point data: E9.0 field (9 columns)

Note that for floating point data entered without a decimal point, the decimal point is assumed to be to the right of the data field.

In preparing data, it should be remembered that all blanks in integer or floating point fields are interpreted as zeros. Therefore, all integers (including exponents of floating point numbers) must be right adjusted.

Card Input and Output

Each physical data card is written on the output file as soon as it is read from the input file. The resulting printout includes the information in card columns (cc) 73 through 80 of the data cards. Since the present version of FASTER does not print details of problem data except for the input cards, prolific use of card labeling is desirable. A note of warning: in obtaining the card identification from cc 73-80, all unused data fields in cc 1-72 are interpreted as data and these unused fields should be blank or contain valid data punches.

Input Data Sections

Data input to FASTER is divided into six sections to simplify the description of multiple change cases. The first data card of each input section is the minimum input requirements. The six sections of data input are:

1) title cards, limits, and options
2) surfaces and regions
3) fixed sources
4) cross sections
5) flux groups, response functions and detectors
6) sampling parameters

Detailed input instructions are given below. To simplify the setup of a problem, it is probably best to start with geometric input data and come back to the data input for section 1 later. This simplifies the necessity of specifying maximum array dimensions before the arrays can be used.

Input Control Procedure

The first data card of each input section contains integer constants controlling the input of the remainder of the data in the section. The general procedure is

input control constant ≤ 0, no input

input control constant > 0, input

When input is present, the control constant may serve a dual purpose by also denoting the quantity of input. As an example, the first input-control constant (IN1) for each data section pertains to hollerith or comment cards. If non-positive (IN1 ≤ 0), no comment cards should be supplied. If desired, any number of comment cards (up to 999) may be inserted immediately after the input control card for each section. The value of the first input-control constant is then set equal tothe number of these comment cards, i.e. IN1 = total number of comment cards.

9.2 SECTION 1 DATA; TITLE CARDS, LIMITS AND OPTIONS

CARD 1-0, Input Controls for Section 1 Data

NOTE: This card is always required.

Column	Format	Symbol	Definition
1-3	I3	IN1	input control for card 1-1 (descriptive cards) omit card 1-1 if IN1 ≤ 0 supply IN1 physical cards if IN1 ≥ 1
4-6	I3	IN2	input control for card 1-2 (first title card) omit card 1-2 if IN2 ≤ 0 supply card 1-2 if IN2 ≥ 1
7-9	I3	IN3	input control for card 1-3 (second title card) omit card 1-3 if IN3 ≤ 0 supply card 1-3 if IN3 ≥ 1
10-12	I3	IN4	input control for card 1-4 (geometric limits) omit card 1-4 if IN4 ≤ 0 supply card 1-4 if IN4 ≥ 1
13-15	I3	IN5	input control for card 1-5 (fixed source limits) omit card 1-5 if IN5 ≤ 0 supply card 1-5 if IN5 ≥ 1
16-18	I3	IN6	input control for card 1-6 (cross section limits) omit card 1-6 if IN6 ≤ 0 supply card 1-6 if IN6 ≥ 1
19-21	I3	IN7	input control for card 1-7 (neutron scattering limits) omit card 1-7 if IN7 ≤ 0 supply card 1-7 if IN7 ≥ 1
22-24	I3	IN8	input control for card 1-8 (flux limits) omit card 1-8 if IN8 ≤ 0 supply card 1-8 if IN8 ≥ 1
25-27	I3	IN9	input control for card 1-9 (sampling options) omit card 1-9 if IN9 ≤ 0 supply card 1-9 if IN9 ≥ 1

Column	Format	Symbol	Definition
28-30	I3	IN10	input control for card 1-10 (iteration limits) omit card 1-10 if IN10 ≤ 0 supply card 1-10 if IN10 ≥ 1
31-33	I3	IN11	printout control for storage map printout no printout of storage map if IN11 ≤ 0 printout of storage map if IN11 ≥ 1
34-72	13I3	---	these columns are not used and should be left blank
73-80	2A4	---	any desired information for card identification

CARD 1-1, Descriptive Information for Section 1 Data

NOTE: a) Omit this card if IN1 ≤ 0
b) Supply IN1 physical cards if IN1 ≥ 1

1-72	18A4	---	any desired information
73-80	2A4	---	any desired information for card identification

CARD 1-2, First Title Card for Labeling the Printout

NOTE: a) Omit this card if IN2 ≤ 0
b) Supply this card if IN2 ≥ 1

1-72	18A4	---	any desired information for identification of the problem; this will then appear on the first line of each printout page
73-80	2A4	---	any desired information for card identification

CARD 1-3, Second Title Card for Labeling the Printout

NOTE: a) Omit this card if IN3 ≤ 0
b) Supply this card if IN3 ≥ 1

1-72	18A4	---	any desired information for identification of the problem; this will appear on the second line of each printout page
73-80	2A4	---	any desired information for card identification

CARD 1-4, Geometric Limits

NOTE: a) Omit this card if IN4 ≤ 0
b) Supply this card if IN4 ≥ 1

Column	Format	Symbol	Definition
1-3	I3	NSMAX	total number of surfaces required for the geometry description
4-6	I3	NAMAX	maximum number of coefficients required to define each surface in the expanded form, e.g. NAMAX = 6 for geometries involving any cones or spheres
7-9	I3	NRMAX	total number of regions required to describe the material distribution including voids
10-12	I3	NBMAX	maximum number of surfaces bounding a region
13-15	I3	NSTMAX	maximum number of regions which can be traversed by a single straight line or ray; the theoretical limit is 2·NSMAX - (number of plane surfaces + 1)
16-72	19I3	---	these columns are not used and should be left blank
73-80	2A4	---	any desired information for card identification

CARD 1-5, Fixed Source Limits

NOTE: a) Omit this card if IN5 ≤ 0
b) Supply this card if IN5 ≥ 1

1-3	I3	NEMAX	number of energy groups used in this problem for source spectra and cross sections; source spectra may be described in a different group structure and are regrouped as noted later

Column	Format	Symbol	Definition
4-6	I3	NVMAX	total number of fixed sources
7-9	I3	NXMAX	maximum number of points used to tabulate each spatial or angular source distribution
10-12	I3	NXEMAX	maximum number of energy points required to tabulate the differential source spectrum; if any source spectra are described by group integrated values, 2 energy points will be generated for each of these groups before the spectrum is integrated into the group structure for the problem.
13-72	20I3	---	these columns are not used and should be left blank
73-80	2A4	---	any desired information for card identification

CARD 1-6, Basic Cross Section Limits and Options

NOTE: a) Omit this card if IN6 $\leq$ 0
b) Supply this card if IN6 $\geq$ 1.

Column	Format	Symbol	Definition
1-3	I3	NXSECT	particle type option. 0 for photons. 1 for neutrons
4-6	I3	NUNITD	composition units option. 0 for 10^{24} atoms/cm^3 1 for gm/cm^3
7-9	I3	NUNITX	microscopic cross sections units option. 0 for barns/atom. 1 for cm^2/gm
10-12	I3	NIMAX	total number of different elements or isotopes
13-15	I3	NMMAX	total number of composite materials; hydrogen densities can be entered by region thus reducing the total number of different compositions, e.g. pure hydrogen compositions need not be defined

Column	Format	Symbol	Definition
16-72	19I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 1-7, Neutron Scattering Limits

NOTE: a) Omit this card if IN7 $\leq$ 0.
b) Supply this card if IN7 $\geq$ 1.
c) For photon problems, define each variable on this card as zero.

Column	Format	Symbol	Definition
1-3	I3	NORDER	Maximum number of elastic scattering transfer matrices for all elements. 1 for P_o (transport approximation) (see notes after card 4-5 for internal cross section juggling) 2 for P_1, 3 for P_2, etc. Negative fluxes can, and will, occur for NORDER > 1 and deep penetrations.
4-6	I3	NDOWN	Maximum group-to-group transfer for elastic scattering for all elements. 1 for in group only, 2 for down 1, etc.
7-9	I3	INELAS	Maximum number of groups for which non-elastic transfer can be initiated for all elements. 0 indicates none if these cross sections are included in the P_o transfer.
10-12	I3	NDOWN	Maximum group-to-group transfer for non-elastic scattering for all elements. 1 for in group only, 2 for down 1, etc.
13-72	20I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 1-8, Flux Limits

NOTE: a) Omit this card if IN8 ≤ 0.
b) Supply this card if IN8 ≥ 1.

Column	Format	Symbol	Definition
1-3	I3	NGMAX	Number of flux groups; less than or equal to the number of groups used for source spectra and cross sections.
4-6	I3	NFMAX	Total number of response functions such as flux-to-dose, energy deposition. 0 indicates no response functions.
7-9	I3	NVMOD	Total number of fixed sources for which separate flux contributions are desired. 0 indicates none.
10-12	I3	NCMAX	Number of separate contributions to the flux by order-of-scatter 0 - none, 1 - uncollided flux, 2- uncollided flux and single scattered flux, etc.
13-15	I3	NLMAX	Number of Legendre moments of the angular flux 0, P_o moment (always obtained, ≡ scalar flux). 1, P_o and P_1 moments, etc.
16-18	I3	NTMAX	Number of length-of-flight moments of the flux. 0, zeroth moment (always obtained, ≡ scalar flux). 1, zeroth and first moments, etc.
19-21	I3	NDMAX	Total number of detectors of all types.
22-24	I3	NSRMAX	Number of regions for which separate scattering contributions are desired.
25-72	16I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 1-9, Random Sampling Options

NOTE: a) Omit this card if IN9 ≤ 0.
b) Supply this card if IN9 ≥ 1.
c) The preferred value of each number on this card is 1.

Column	Format	Symbol	Definition
1-3	I3	NPOINT	0, calculate fluxes for all detectors simultaneously; the preferred point is defined by input, point detectors must be in void regions. 1, calculate fluxes for each point detector individually; the preferred point(s) is the point detector - surface and volume detectors are ignored.
4-6	I3	MODELP	0, randomly select the fixed source and then randomly select the spatial source variables in the source coordinate system. 1, randomly select the spatial source variables in a spherical coordinate system centered at the preferred point; all sources must be volumetric; source volumes must completely cover one or more regions.
7-9	I3	MODELQ	0, randomly select angular source variables in the source coordinate system. 1, randomly select angular source variables like MODELV below, with the direction-before-scattering defined as a unit vector from the center of all sources to the selected source point; all sources must be angular or isotropic.
10-12	I3	MODELU	Distance between scatters random selection option. 0, exponential transformation. 1, curve fit of optimum function (difficulties may be encountered for large volumes since the curve fit is from boundary to boundary).
13-15	I3	MODELV	Direction vector random selection option. 0, polar angle measured from direction before scattering; azimuthal angle measured from a unit direction vector towards the preferred point. 1, opposite of the above. 2 and 3, polar angle measured from unit vector towards preferred point with combined importance parameters from equations 7.42 and 7.45, or 7.41 and 7.45 respectively; azimuthal angles equiprobable.

Column	Format	Symbol	Definition
16-72	19I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 1-10, Iteration Limits

NOTE: a) Omit this card if IN10 ≤ 0.
b) Supply this card if IN10 ≥ 1.

Column	Format	Symbol	Definition
1-3	I3	NPRINT	Total number of printouts during the flux calculations; yields a convergence history and protects against complete loss if the problem is terminated.
4-6	I3	NUNITS	Number of outer iterations between printouts, i.e., the number of packets of particles.
7-9	I3	NUMBER	Number of discrete directions obtained by random sampling used in integrating the angular point sources to obtain surface and volume averaged fluxes; not used if NPOINT = 1, or if all detectors are points.
10-12	I3	KALIDE	Maximum number of inner iterations per outer iteration, i.e. the number collisions per packet.
13-72	20I3	---	These columns are not used and should be left blank.
73-78	2A4	---	Any desired information for card identification.

9.3 SECTION 2 DATA; SURFACES AND REGIONS

CARD 2-0, Input Controls for Section 2 Data

NOTE: a) This card is always required.

Column	Format	Symbol	Definition
1-3	I3	IN1	Input control for Card 2-1 (descriptive cards). Omit Card 2-1 if IN1 ≤ 0. Supply IN1 physical cards if IN1 ≥ 1.
4-6	I3	IN2	Input control for Card 2-2 (surfaces). Omit Card 2-2 if IN2 ≤ 0. Supply IN2 physical cards if IN2 ≥ 1, i.e. IN2 surfaces will be described.
7-9	I3	IN3	Input control for Card 2-3 (regions) Omit Card 2-3 if IN3 ≤ 0. Supply IN3 physical cards if IN3 ≥ 1, i.e. IN3 regions will be described.
10-12	I3	IN4(1)	Ambiguity index calculation option. 0, do not calculate ambiguity indices. 1, calculate ambiguity indices. Use IN4 = 1 on the first problem.
13-15	I3	IN5(1)	Geometry consistency check option. 0, do not check geometry. 1, check geometry. Use IN5 = 1 on the first problem.
16-72	19I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

(1) IN4 = IN5 = 0 permits the redefinition of regions by surface description changes only. Extreme caution should be used to ensure that the ambiguity indices, calculated in previous problems, are still correct.

CARD 2-1, Description of Section 2 Data

NOTE: a) Omit this card if IN1 ≤ 0.
b) Supply IN1 physical cards if IN1 ≥ 1.

Column	Format	Symbol	Definition
1-72	18A5	---	Any desired information for description of the input data.
73-80	2A4	---	Any desired information for card identification.

CARD 2-2, Surface Description

NOTE: a) Omit this card if IN2 ≤ 0.
b) Supply IN2 physical cards if IN2 ≥ 1.

Column	Format	Symbol	Definition
1-3	I3	I	Index of the surface being described.
4-6	I3	NTP(I)	Index (j) of the last non-zero coefficient if the surface is in the expanded form; calculated internally for all other surfaces.
7-9	I3	NEX	Form (n_x) of the surface as input. 0, already in expanded form. 1, ≤ NEX ≤13, special form as indicated in figures 7, 8, and 9 and table 1.
10-18	E9.0	AA(1)	First parameter defining the surface.
19-27	E9.0	AA(2)	Second parameter defining the surface.
⋮	⋮	⋮	⋮
64-72	E9.0	AA(7)	Seventh parameter defining the surface.
73-80	2A4	---	Any desired information for card identification.

The requisite parameters are listed in the last column of table 1 and are input in the order shown. If the surface is in the expanded form and rotational terms are involved, supply these on Card 2-2' before supplying Card 2-2 for the next input surface.

TABLE 1

SPECIAL SURFACE EQUATIONS

Surface Type	Surface Equation	Last Term	Type n_x	Input
quadric	$a_0+a_1x+a_2y+a_3z+a_4x^2+a_5y^2+a_6z^2+a_7xy+a_8yz+a_9zx = 0$	i (input)	0	$a_0, a_1, \ldots, a_i$
plane ⊥ x-axis	$x = c$	1	1	c
plane ⊥ y-axis	$y = c$	2	2	c
plane ⊥ z-axis	$z = c$	3	3	c
plane ∥ x-axis	$(y-y_0)/(z-z_0) = (y_1-y_0)/(z_1-z_0)$	3	4	y_0, z_0, y_1, z_1
plane ∥ y-axis	$(x-x_0)/(z-z_0) = (x_1-x_0)/(z_1-z_0)$	3	5	x_0, z_0, x_1, z_1
plane ∥ z-axis	$(x-x_0)/(y-y_0) = (x_1-x_0)/(y_1-y_0)$	2	6	x_0, y_0, x_1, y_1
cone ∥ x-axis	$\left(\left[(y-y_0)^2+(z-z_0)^2\right]^{1/2}-r_0\right)/(x-x_0) = (r_1-r_0)/(x_1-x_0)$	6	7	$y_0, z_0, r_0, x_0, r_1, x_1$
cone ∥ y-axis	$\left(\left[(x-x_0)^2+(z-z_0)^2\right]^{1/2}-r_0\right)/(y-y_0) = (r_1-r_0)/(y_1-y_0)$	6	8	$x_0, z_0, r_0, y_0, r_1, y_1$
cone ∥ z-axis	$\left(\left[(x-x_0)^2+(y-y_0)^2\right]^{1/2}-r_0\right)/(z-z_0) = (r_1-r_0)/(z_1-z_0)$	6	9	$x_0, y_0, r_0, z_0, r_1, z_1$
cylinder ∥ x-axis	$(y-y_0)^2/a^2 + (z-z_0)^2/b^2 = 1$	6	10	y_0, a, z_0, b
cylinder ∥ y-axis	$(x-x_0)^2/a^2 + (z-z_0)^2/b^2 = 1$	6	11	x_0, a, z_0, b
cylinder ∥ z-axis	$(x-x_0)^2/a^2 + (y-y_0)^2/b^2 = 1$	5	12	x_0, a, y_0, b
ellipsoid	$(x-x_0)^2/a^2 + (y-y_0)^2/b^2 + (z-z_0)^2/c^2 = 1$	6	13	x_0, a, y_0, b, z_0, c

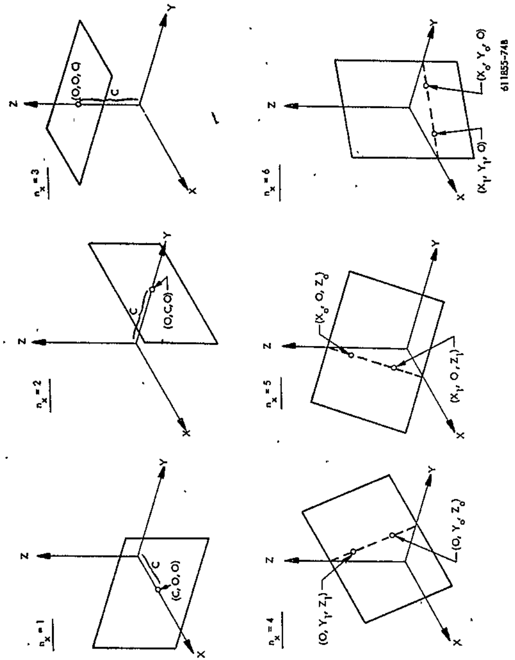


Figure 7. Plane Surfaces

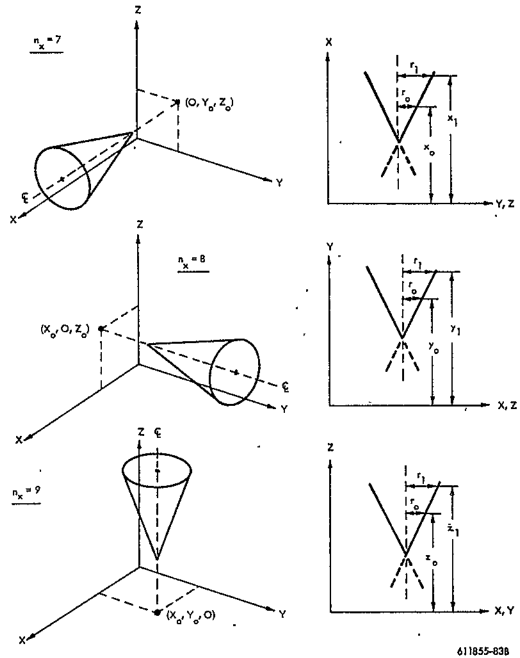


Figure 8. Conical Surfaces

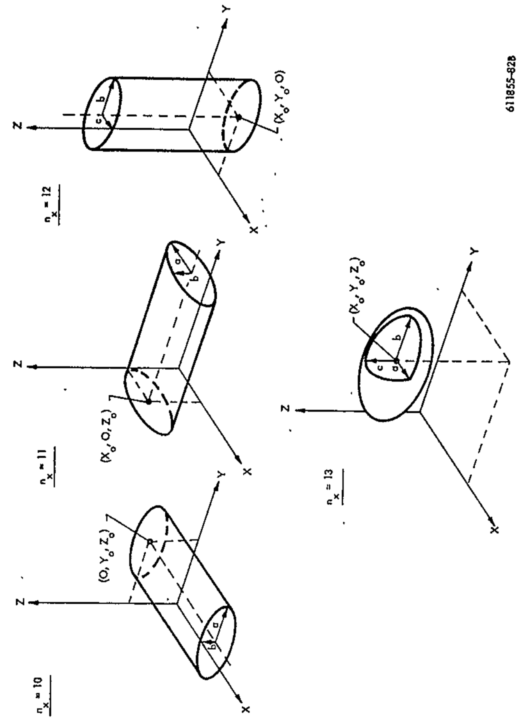


Figure 9. Elliptical Cylinders and Ellipsoids

CARD 2-2', Rotated Surface Equation Terms

NOTE: a) Supply this card as required to finish the description of a surface, omit otherwise.

Column	Format	Symbol	Definition
1-9	E9.0	A(7, I)	Coefficient of xy in the general surface equation.
10-18	E9.0	A(8, I)	Coefficient of yz in the general surface equation.
19-27	E9.0	A(9, I)	Coefficient of zx in the general surface equation.
28-72	5E9.0	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 2-3, Region Description

NOTE: a) Omit this card if IN3 ≤ 0.
b) Supply IN3 physical cards if IN3 ≥ 1.

1-3	I3	I	Index of the region being described.
4-6	I3	ISV(I)	Index of the volume source superimposed over this region. 0, indicates none. Required if and only if MODELP = 1. May also be input or changed on Card 3-6.
7-9	I3	MTL(I)	> 0, index of the composition for the region. =0, the region contains hydrogen only. < 0, the region is void. This index can be input or changed on Card 4-8.
10-12	I3	NS(1, I)	First boundary surface index.
13-15	I3	NS(2, I) ⋮	Second boundary surface index. 0 or blank if all boundaries have been listed.

Column	Format	Symbol	Definition
		⋮	
	I3	NS(J, I)	Jth boundary surface index. 0 or blank if all boundaries have been listed.
		⋮	
34-36	I3	NS(9, I)	> 0, ninth boundary surface index if the region has exactly nine boundaries. 0, or blank if all boundaries have been listed (less than nine boundaries). -1, if the region has more than nine boundaries; the ninth and remaining boundaries are listed on Card 2-3'.
37-45	E9.0	RHO(I)	Hydrogen density in the region apart from that specified for the composite material in the region; units according to NUNITD, may be input or changed on Card 4-9.
46-54	E9.0	XR(1, I)	x-coordinate of any point in the region (cm).
55-63	E9.0	XR(2, I)	y-coordinate of the point in the region (cm).
64-72	E9.0	XR(3, I)	z-coordinate of the point in the region (cm).
73-80	2A4	---	Any desired information for card identification.

CARD 2-3', Additional Region Boundaries

NOTE: a) Supply this card(s) for each region having more than nine boundaries, immediately behind Card 2-3 for the region; omit otherwise.
b) This card contains data up to and including the maximum number of boundaries (more than 1 physical card if NBMAX > 32).

Column	Format	Symbol	Definition
1-72	24I3	NS(9, I)	Ninth boundary surface index.
		NS(10, I)	Tenth boundary surface index.
		NS(11, I)	Eleventh boundary surface index. 0 or blank if all boundaries are listed.
		⋮	

Column	Format	Symbol	Definition
		NS(NBMAX, I)	Maximum boundary surface index. 0 or blank if all boundaries are listed.
73-80	2A4	---	Any desired information for card identification.

9.4 SECTION 3 DATA; SOURCE DISTRIBUTIONS

CARD 3-0, Input Controls for Section 3 Data

NOTE: a) This card is always required.

Column	Format	Symbol	Definition
1-3	I3	IN1	Input control for Card 3-1 (descriptive cards). Omit Card 3-1 if IN1 ≤ 0. Supply IN1 physical card if IN1 ≥ 1.
4-6	I3	IN2	Input control for Card 3-2 (energy levels). Omit Card 3-2 if IN2 ≤ 0. Supply Card 3-2 if IN2 ≥ 1.
7-9	I3	IN3	Input control for fixed sources, omit Cards 3-3 through 3-5 if IN3 ≤ 0. Supply Cards 3-3, 3-4 and 3-5 as required for IN3 fixed sources if IN3 ≥ 1.
10-12	I3	IN4	Input control for Card 3-6 (source-in-region). Omit Card 3-6 if IN4 ≤ 0. Supply source in region indices on Card 3-6 for IN4 regions if IN4 ≥ 1.
13-72	20I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 3-1, Description of Section 3 Data

NOTE: a) Omit this card if IN1 ≤ 0.
b) Supply IN1 physical cards if IN1 ≥ 1.

Column	Format	Symbol	Definition
1-72	18A4	---	Any desired information for describing the input data.
73-80	2A4	---	Any desired information for card description.

CARD 3-2, Energy Levels for Sources and Cross-Sections

Column	Format	Symbol	Definition

NOTE: a) Omit this card if IN2 ≤ 0.

b) Supply this card(s) if IN2 ≥ 1.

c) Source spectra can be input in any desired group structure and will be regrouped to this set of groups.

Column	Format	Symbol	Definition
1-72	8E9.0	ELL(1)	Upper energy boundary of the first energy group (Mev).
		ELL(2)	Lower energy boundary of the first energy group and upper boundary of the second group
		• • •	
		ELL(NEMAX+1)	Lower energy boundary of the last energy group; also defines the energy cutoff point.
72-80	2A4	----	Any desired information for card identification.

Card 3-3, Fixed Source Constants and Input Options

NOTE: a) Omit this card if IN3 ≤ 0.

b) Supply this card, and Cards 3-4 and 3-5 as required, for IN3 sources if IN3 ≥ 1.

Column	Format	Symbol	Definition
1-3	I3	I	Index of source being described.
4-6	I3	NSG(I)	Source geometry type. 0, rectangular 1, cylindrical 2, spherical
7-9	I3	NPC(1, I)	First spatial variable distribution option. (See notes below.)
10-12	I3	NPC(2, I)	Second spatial variable distribution option. (See notes below.)
13-15	I3	NPC(3, I)	Third spatial variable distribution option. (See notes below.)

Column	Format	Symbol	Definition
16-18	I3	NPC(4, I)	Azimuthal part of angular distribution option (fourth source variable. (See notes below.)
19-21	I3	NPC(5, I)	Polar part of the angular distribution, option (fifth source variable). (See notes below.)

NOTES: The source variables are shown in Figure 10 and are ordered as:

	Rectangular	Cylindrical	Spherical
J = 1	x (cm)	r (cm)	ρ (cm)
J = 2	y (cm)	θ (radians)	θ (radians)
J = 3	z (cm)	z (cm)	μ
J = 4	θ' (radians)	θ' (radians)	θ' (radians)
J = 5	μ'	μ'	μ'

Azimuthal angles are in the range $-\pi \leq \theta,\ \theta' \leq \pi$

Cosines of polar angles are in the range $-1 \leq \mu,\ \mu' \leq 1$

If NPC (J, I) > 0, this is the number of tabulation points required to describe the Jth distribution using Card 3-4. If NPC(J, I) < 0, the distribution for variable J of source I' = -NPC(J, I) is used and Card 3-4 is not required.

Column	Format	Symbol	Definition
22-24	I3	MAX	> 0, number of energy points or energy groups required to describe the source spectrum < 0, use the source spectrum for source number I'=-MAX.
25-27	I3	NORM	Spectrum normalization option (the total source strength is carried in the spectrum). 0, normalize to total source in particles/sec; 1, normalize to total source in Mev/sec; 2, multiply spectrum by constant (if used with MAX < 0, remember that the spectrum for source I' = -MAX has been normalized to the total source strength).

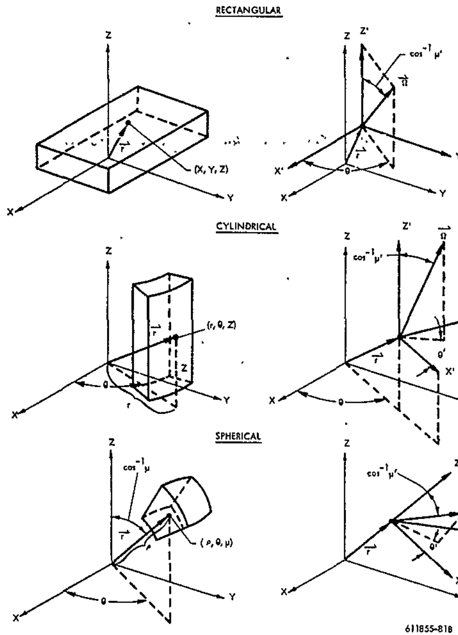


Figure 10. Source Distribution Variables

Astronuclear Laboratory

Column	Format	Symbol	Definition
28-30	I3	ISP	Input spectrum units option if MAX > 0 0, differential number spectrum at energy points; 1, differential energy spectrum at energy points; 2, groupwise number spectrum (particles in group); 3, groupwise energy spectrum (energy in group).
31-33	I3	IAL	Spectrum format option if MAX > 0 0, alternating values of energy points and spectrum 1, spectrum input only using energy points previously input for this case 2, energy point input followed by spectrum input on separate card
34-36	I3	----	These columns are not used and should be left blank.
37-45	E9.0	TOT	Source normalization constant, particles/sec if NORM = 0, Mev/sec if NORM = 1, multiplying constant if NORM = 2.
46-54	E9.0	XTR(1, I)	x component of the translation of the source coordinate system origin from the geometry coordinate system origin (cm).
55-63	E9.0	XTR(2, I)	y component of the source translation (cm).
64-72	E9.0	XTR(3, I)	z component of the source translation (cm).
73-80	2A4	----	Any desired information for card identification.

Card 3-4, Source Variable Distributions

NOTE:
a) Supply these cards immediately behind Card 3-3 for the source to which they apply. Input for each variable in the order J = 1, 2, 3, 4, 5, starting a new physical card for each variable.
b) Omit for any variable for which NPC(J, I) < 0.
c) This distribution is normalized and interpolated linearly.
d) Histogram data can be used (points can coincide).

Column	Format	Symbol	Definition
1-72	8E9.0	VEE(1,J,I)	Minimum value of the Jth variable; if NPC(J,I) = 1, this is the only value of the variable,
		VAL(1,J,I)	*Relative value of the distribution function* for the Jth variable at its minimum value (not used if the variable is discrete, NPC(J,I) = 1)
		• • •	
		VEE(K,J,I)	Kth value of the Jth variable.
		VAL(K,J,I)	Relative value of the distribution function corresponding to VEE(K,J,I).
		• • •	
		VEE(L,J,I)	Maximum value of the Jth variable where L=NPC(J,I), the last tabulation point.
		VAL(L,J,I)	Relative value of the distribution function corresponding to VEE(L,J,I).
73-80	2A4	----	Any desired information for card identification.

NOTE: To avoid numerical difficulties, it is sometimes necessary to decrement the minimum value and increment the maximum value of these variables. In particular, for a uniform azimuthal distribution angular limits use 3.1416 ($>\pi$) for π rather than 3.14159 ($<\pi$) since the computer uses more significant figures than can be input.

CARD 3-5, Source Spectrum

NOTE: a) Supply this card(s) if and only if MAX > 0. It should be placed immediately after the source variable distributions, if any.

Column	Format	Symbol	Definition
1-72	8E9.0	----	See notes below.
73-80	2A4	----	Any desired information for card identification.

NOTES: A. If ISP $\leq$ 1, the differential spectrum is tabulated at discrete energy points where
E(1) is the maximum spectral energy (Mev)
•
•
•
E(MAX) is the minimum spectral energy
EN(K) is the differential spectrum corresponding to the Kth energy point E(K). The units of EN(K) are particles/Mev if ISP = 0 or Mev/Mev if ISP = 1.

A.1 If IAL = 0, the input on Card 3-5 consists of alternating values of energy *and spectrum*

E(1), EN(1), E(2), EN(2), . . . , E(MAX), EN(MAX)

A.2 If IAL = 1, the energy points are already defined by prior input for this case (they will not be available from a previous case). The input consists of the relative spectrum at these points

EN(1), EN(2), . . . , EN(MAX)

A.3 If IAL = 2, the energy points are defined first

E(1), E(2), . . . , E(MAX)

and then followed by another card with the corresponding spectrum

EN(1), EN(2), . . . , EN(MAX)

B. If ISP $\geq$2, a groupwise integrated spectrum is tabulated by group where
EBG(1) is the upper energy boundary of group 1
•
•
•
EBG(MAX+1) is the lower energy boundary of the last spectrum group
ENG(K) is the integral spectrum for the Kth group with units of particles in group K if ISP=2, or Mev in group K if ISP = 3.

Column	Format	Symbol	Distribution

B. 1 If IAL = 0, the input on Card 3-5 consists of alternating values of energy group boundaries and group spectrum.
EBG(1), ENG(1), EBG(2), ENG(2), . . . , EBG(MAX), ENG(MAX, EBG(MAX+1)

B. 2 If IAL = 1, the energy group boundaries are already defined and the groupwise spectrum is supplied on Card 3-5
ENG(1), ENG(2), . . . , ENG(MAX)

B. 3 If IAL = 2, the energy group boundaries are defined first
EBG(1), EBG(2), . . ., EBG(MAX+1)
and then followed by another card with the groupwise spectrum
ENG(1), ENG(2), . . ., ENG(MAX)

CARD 3-6, Source in Region Indices

NOTE: a) Omit this card if IN4 ≤ 0.
b) Supply this card(s) if IN4 ≥ 1 with source indices for IN4 regions.
c) This data is required if and only if MODELP=1.
d) This data can also be input on Card 2-3.

Column	Format	Symbol	Distribution	
1-72	24I3	I	First geometric region index	K = 1
		ISV(I)	Index of source superimposed over Region I (source which completely covers the region), 0 denotes none	K = 1
		I	Second geometric region index	K = 2
		ISV(I)	Index of source superimposed over Region I (source which completely covers the region)	K = 2
		⋮		⋮
		I	Last geometric region index	K = IN4
		ISV(I)	Index of source superimposed over Region I (source which completely covers the region)	K = IN4
73-80	2A4		Any desired information for card identification.	

9.5 SECTION 4 DATA, MICROSCOPIC CROSS SECTIONS

CARD 4-0, Input Controls for Section 4 Data

NOTE: a) This card is always required.

Column	Format	Symbol	Definition
1-3	I3	IN1	Input control for Card 4-1 (descriptive cards) Omit Card 4-1 if IN1 ≤ 0. Supply IN1 physical cards if IN1 ≥ 1
4-6	I3	IN2	Input control for Card 4-2 (neutron scattered energies). Omit Card 4-2 if IN2 ≤ 0. Supply Card 4-2 if IN2 ≥ 1.
7-9	I3	IN3	Input control for microscopic cross sections; Omit Cards 4-3 through 4-7 if IN3 ≤ 0; Supply Cards 4-3 through 4-7 as required if IN3 ≥ 1.
10-12	I3	IN4	Input control for Card 4-8 (material-in-region) Omit Card 4-8 if IN4 ≤ 0. Supply Card 4-8 with IN4 material-in-region indices if IN4 ≥ 1.
13-15	I3	IN5	Input control for Card 4-9 (hydrogen in region) Omit Card 4-9 if IN5 ≤ 0. Supply Card 4-9 with IN5 hydrogen-in-region densities if IN5 ≥ 1.
16-18	I3	IN6	Cross section output option (used only if IN3 ≥ 1), no output if IN6 ≤ 0. If IN6 ≥ 1, total cross sections are printed by group (neutrons) or level (photons) and by material. Heating responses are printed by energy level for each element in (Mev/atom/unit number flux) x 10^{24} and then for each composite material in (Mev/cm^3/unit number flux).

Column	Format	Symbol	Definition
19-72	18I3	----	These columns are not used and should be left blank.
73-80	2A4	----	Any desired information for card identification.

CARD 4-1, Descriptive Information for Section 4 Data

NOTE: a) Omit this card if IN1 ≤0.
b) Supply IN1 physical cards if IN1 ≥1.

Column	Format	Symbol	Definition
1-72	18A4	----	Any desired information for description of the input data.
73-80	2A4	----	Any desired information for card identification.

CARD 4-2, Average Neutron Energies After Scatter

NOTE: a) Omit this card if IN2 ≤0.
b) Supply this card (s) if IN2 ≥1.
c) This data is required if and only if NXSECT=1 (neutron problem)

Column	Format	Symbol	Definition
1-72	8E9.0	ESB(1)	Average neutron energy for group 1 (Mev)
		ESB(2)	Average neutron energy for group 2
		• • •	
		ESB(NEMAX)	Average neutron energy for the last energy group.
73-80	2A4	----	Any desired information for card identification.

CARD 4-3, Composition Vector for Element

NOTE: a) Omit Cards 4-3 through 4-7 if IN3 ≤0.
b) Supply Cards 4-3 through 4-7 as required in sets for each element, i.e. all data for the first element, all data for the second element, etc., through the data for element number NIMAX if IN3 ≥1.
c) The first element must always be hydrogen.

Column	Format	Symbol	Definition
1-72	8E9.0	ATW	Atomic weight of the element in atomic mass units.
		ZEE	Atomic number of the element.
		ATD(1)	Density of the element in composite material1, units according to NUNITD.
		ATD(2)	Density of the element in composite material 2.
		• • •	
		ATD(NMMAX)	Density of the element in the last composite material.
73-80	2A4	----	Any desired information for card identification.

CARD 4-4, Microscopic Total Cross Sections

NOTE: a) Supply this card immediately after Card 4-3 for each element.

Column	Format	Symbol	Definition
1-72	8E9.0	XST(1)	Total microscopic cross section for energy level 1 (photons) or energy group 1 (neutrons), units according to NUNITX.
		XST(2)	Total microscopic cross section for energy level 2 or energy group 2.
		• • •	
		XST(NEMAX)	Total microscopic cross section for next-to-last photon energy level or last neutron energy group.
		XST(NEMAX+1)	Total microscopic photon cross section for the last energy level. Do not input a number for neutrons.
73-80	2A4	----	Any desired information for card identification.

CARD 4-5, Neutron Transfer Cross Section Array Limits

NOTE: a) Omit this card for photon problems.
b) Omit this card for the first element (always hydrogen) of neutron problems.
c) Supply this card after Card 4-4, for all elements except the first of neutron problems.

Column	Format	Symbol	Definition
1-72	24I3	LMAX	|LMAX|, number of elastic scattering transfer matrices. 1 for P_o only 2 for P_o and P_1, etc. LMAX < 0, indicates total cross sections are transport corrected. (See notes below.)
		NDSM	Maximum group-to-group transfer for elastic scattering 1 for in-group only 2 for down 1, etc.
		JMAX	Maximum number of groups for which non-elastic transfer can be initiated. 0, none
		KMX(1)	Maximum non-elastic group-to-group transfer for initial group 1, = 1 for in group only, etc.
		KMX(2)	Maximum non-elastic group-to-group transfer for initial group 2, = 1 for in group only, etc.
		KMX(JMAX)	Maximum non-elastic group-to-group transfer for the last possible initial group, = 1 for in group only, etc.
73-80	2A4	----	Any desired information for card identification.

Astronuclear Laboratory

NORDER = 1, LMAX = -2, Code transport corrects P_o elastic transfer using P_1 transfer

NORDER = 1, LMAX = +2, Code transport corrects P_o elastic transfer and calculates transport cross sections using P_1 transfer.

NORDER ≥ 2, LMAX = -2, Code calculates total cross section using P_1 transfer.

NORDER = 1, LMAX = ± 1, Cross sections used as input.

NORDER ≥ 2, LMAX ≥ 2, Cross sections used as input.

CARD 4-6, Neutron Elastic Transfer Coefficients

NOTE: a) Supply this card for all elements except the first of neutron problems immediately after Card 4-5.

Column	Format	Symbol	Definition
1-72	8E9.0	$\sigma^l_{j \to k}$ XSE(J, K, L)	(1) lth Legendre moment of the transfer cross section from group j to k including (2l + 1) coefficient, e.g., GAM-1, GAM-2 printed output.
73-80	2A4	----	Any desired information for card identification.

(1) Start a card with P_o in-group transfer for all energy groups

$\sigma^o_{i \to i}$, i = 1, 2, ... , NEMAX

Start a new card with P_o down 1 transfer for all groups except the last

$\sigma^{o}_{i \rightarrow i+1}$, i = 1, 2, ... , NEMAX - 1

•
•
•

Start a new card with P_o down (NDSM-1) transfer

$\sigma^{o}_{i \rightarrow i + NDSM - 1}$, i = 1, 2, ... , NEMAX - (NDSM-1)

Start a new card with P_1 in group transfer

$\sigma^{1}_{i \rightarrow i}$, i = 1, 2, ... , NEMAX

•
•
•

Start a new card with $P_{|LMAX| - 1}$ down (NDSM-1) transfer

$\sigma^{|LMAX| - 1}_{i \rightarrow i + NDSM-1}$ i = 1, 2, ..., NEMAX - (NDSM-1)

CARD 4-7, Neutron Non-Elastic Transfer Coefficients

NOTE: a) Supply this card for all elements--except the first--of neutron problems immediately after Card 4-6.
b) Omit if JMAX = 0.

Column	Format	Symbol	Definition
1-72	8E9.0	$\sigma^{ne}_{i \rightarrow i + k - 1}$	(1)Non-elastic transfer coefficient from
		XSI(K, J)	group j to group j + k - 1
			$= \sigma^{inelastic}_{i \rightarrow i + K - 1} + 2\, \sigma^{n-2n}_{i \rightarrow i + k - 1} + \ldots$
73-80	2A4	----	Any desired information for card identification.

(1) Start the first card with non-elastic transfer from group 1

$\sigma^{ne}_{1 \rightarrow 1 + K - 1}$ K = 1, 2, ..., KMX(1)

Start a new card with non-elastic transfer from group 2

$\sigma^{ne}_{2 \rightarrow 2 + K - 1}$, K = 1, 2, ... , KMX(2)

•
•
•

Start a new card with non-elastic transfer from group JMAX

$\sigma^{ne}_{JMAX \rightarrow JMAX + K - 1}$, K = 1, 2, ... , KMX (JMAX)

CARD 4-8, Material-In-Region Indices

NOTE: a) Omit this card if IN4 ≤ 0.
b) Supply this card if IN4 ≥ 1 with material indices for IN4 regions.
c) These indices can also be input on Card 2-3.

Column	Format	Symbol	Definition	
1-72	24I3	I	First region index	K=1
		MTL(I)	>0, index of material in this region	
			= 0, region contains hydrogen only	
			< 0, region is void	
		I	Second region index	K=2
		MTL(I)	Material index for region	
		• • •		• • •
		I	Last region index	K=IN4
		MTL(I)	Material Index for region	

Column	Format	Symbol	Definition
73-80	2A4	----	Any desired information for card identification.

CARD 4-9, Hydrogen Density in Region

NOTE: a) Omit this card if IN5 $\leq$ 0.
b) Supply this card if IN5 $\geq$ 1 with densities for IN5 regions.
c) These densities can also be entered on Card 2-3.

Column	Format	Symbol	Definition	
1-72	6(I3, E9.0)	I	First Region Index	K=1
		RHO(I)	Hydrogen density in region I units according to NUNITD	K=1
		I	Second region index	K=2
		RHO(I)	Hydrogen density in region I	K=2
		⋮		⋮
		I	Last region index	K=IN5
		RHO(I)	Hydrogen density in region I	K=IN5
73-80	2A4	----	Any desired information for card identification.	

9.6 SECTION 5 DATA; DETECTORS AND FLUX CONVERSIONS

CARD 5-0, Input Controls for Section 5 Data

NOTE: a) This card is always required.

Column	Format	Symbol	Definition
1-3	I3	IN1	Input control for Card 5-1 (descriptive cards) Omit Card 5-1 if IN1 $\leq$ 0. Supply IN1 physical cards if IN1 $\geq$ 1.
4-6	I3	IN2	Input control for Card 5-2 (flux groups) Omit card 5-2 if IN2 $\leq$ 0. Supply Card 5-2 if IN2 $\geq$ 1.

Column	Format	Symbol	Definition
7-9	I3	IN3	Input control for Card 5-3 (response functions) Omit Card 5-3 if IN3 $\leq$ 0. Supply IN3 response functions if IN3 $\geq$ 1.
10-12	I3	IN4	Input control for Card 5-4 (detectors) Omit Card 5-4 if IN4 $\leq$ 0. Supply IN4 detectos if IN4 $\geq$ 1.
13-15	I3	IN5	Input control for Card 5-5 (Δ total flux sources) Omit Card 5-5 if IN5 $\leq$ 0. Supply Card 5-5 if IN5 $\geq$ 1.
16-18	I3	IN6	Input control for Card 5-6 (scattered flux regions) Omit Card 5-6 if IN6 $\leq$ 0 Supply Card 5-6 if IN6 $\geq$ 1
19-72	18I3	----	These columns are not used and should be left blank.
73-80	2A4	----	Any desired information for card identification.

CARD 5-1, Descriptive Information for Section 5 Data

NOTE: a) Omit this card if IN1 $\leq$ 0.
b) Supply IN1 physical cards if IN1 $\geq$ 1.

Column	Format	Symbol	Definition
1-72	18A4	----	Any desired information for describing the input data.
73-80	2A4	----	Any desired information for card identification.

CARD 5-2, Flux Groups

NOTE: a) Omit this card if IN2 $\leq$ 0.
b) Supply this card if IN2 $\geq$ 1.
c) This card is required if the number of flux groups (NGMAX) is less than the number of groups (NEMAX) used to run the problem.

Column	Format	Symbol	Definition
1-72	24I3	NTG(1)	Index of flux group corresponding to first source and cross section group = 1.

Column	Format	Symbol	Definition
		NTG(2)	Index of flux group corresponding to the second source and cross section group.
		• • •	
		NTG(NEMAX)	Index of flux group corresponding to the last source and cross section group, = NGMAX.
73-80	2A4	---	Any desired information for card identification.

CARD 5-3, Response Functions

NOTE: a) Omit this card if IN3 ≤ 0.
b) Supply this card for IN3 response functions if IN3 ≥ 1.

Column	Format	Symbol	Definition
1-3	I3	I	Index of the response function.
4-6	I3	NTP	Response function type. 0, number flux response input by flux group boundary with units (response/particle cm^{-2} sec^{-1}). 1, energy flux response input by flux group boundary with units (response/Mev cm^{-2} sec^{-1}) <0, energy deposition response function for region I' = -NTP with units (Mev cm^{-3} sec^{-1}/particle cm^{-2} sec^{-1}). NTP <0 requires input of microscopic cross sections in Section 4 data for this problem. (Requires no other input after Column 27 of this card. The response function will appear on the printout immediately below the data on this card.)
7-18	3A4	---	Any desired description of the response function used in labeling the output.
19-27	E9.0	FST	Response function scaling factor, this multiplicative factor can be used to convert the response units to more useful units. Do not use FST = 0.0.
28-36	E9.0	RSP(1, I)	Response function for upper boundary of the first flux group. Leave blank if NTP <0.
		• • •	

Column	Format	Symbol	Definition
63-72	E9.0	RSP(5, I)	Response function for the upper boundary of the 5th flux group (lower boundary of the 4th flux group). Continue on card 5-3' if more than 4 flux groups and NTP ≥ 0. Leave blank if NTP < 0.
73-80	2A4	---	Any desired information for card identification.

CARD 5-3', Response Function Continuation

NOTE: Supply this card immediately behind Card 5-3 if there are more than 4 flux groups and if NTP ≥ 0.

Column	Format	Symbol	Definition
1-72	8E9.0	RSP(6, I)	Response function for the lower boundary of the 5th flux group (upper boundary of the 6th group).
		• • •	
		RSP(NGMAX+1, I)	Response function for the lower boundary of the last flux group.
73-80	2A4	---	Any desired information for card identification.

CARD 5-4, Detectors

NOTE: a) Omit this card if IN4 ≤ 0.
b) Supply IN4 physical cards if IN4 ≥ 1.

Column	Format	Symbol	Definition
1-3	I3	I	Index of the detector being described.
4-6	I3	IDR(I)	0, for point dector >0, region index for a surface or volume detector.
7-9	I3	IDS(I)	0 for a point detector. 0 for a volume detector. >0, surface index for a surface detector (the detector is that part of surface IDS(I) which bounds region IDR(I)).
10-18	E9.0	VOL(I)	Not used for point detector, region volume (cm^3) for volume detector; detector area (cm^2) for surface detector (1.0 yields surface or volume integrated fluxes).

Column	Format	Symbol	Definition
19-27	E9.0	CDT(1, I)	Relative direction cosine with respect to the x - axis of the unit direction vector used in obtaining Legendre moments of the angular flux (not used for surface detectors, angular moments are obtained with respect to the surface normal).
28-36	E9.0	CDT(2, I)	Relative direction cosine with respect to the y - axis.
37-45	E9.0	CDT(3, I)	Relative direction cosine with respect to the z - axis, the 3 direction cosines are normalized by the program.
46-54	E9.0	XDT(1, I)	x coordinate if a point detector (cm).
55-63	E9.0	XDT(2, I)	y coordinate if a point detector (cm).
64-72	E9.0	XDT(3, I)	z coordinate if a point detector (cm).
73-80	2A4	---	Any desired information for card identification.

CARD 5-5, Flux Contribution Sources

NOTE: a) Omit this card if IN5 ≤ 0.
b) Supply this card if IN5 ≥ 1.

Column	Format	Symbol	Definition
1-72	24I3	ISV(1)	Index of first source for which the individual scalar flux contribution is required.
		⋮	
		ISV(NVMOD)	Index of the last source for which the individual scalar flux contribution is required.
73-80	2A4	---	Any desired information for card identification.

CARD 5-6, Scattered Flux by Scattering Region

NOTE: a) Omit this card if IN6 ≤ 0
b) Supply this card if IN6 ≥ 1

Column	Format	Symbol	Definition
1-72	24I3	ISR(1)	Index of the first non-void region from which the individual scattered scalar flux contribution is required.
		⋮	
		ISR(NSRMAX)	Index of the last non-void region from which the scattered flux contribution is required.
73-80	2A4	---	Any desired information for card identification.

9.7 SECTION 6 DATA; RANDOM SAMPLING PARAMETERS

CARD 6-0, Input Controls for Section 6 Data

NOTE: This card is always required.

Column	Format	Symbol	Definition
1-3	I3	IN1	Input control for Card 6-1 (descriptive cards) Omit Card 6-1 if IN1 ≤ 0. Supply IN1 physical cards if IN1 ≥ 1.
4-6	I3	IN2	Input control for Card 6-2 (spherical source and detector). Omit Card 6-2 if IN2 ≤ 0. Supply Card 6-2 if IN2 ≥ 1.
7-9	I3	IN3	Input control for Card 6-3 (source importance). Omit Card 6-3 if IN2 ≤ 0. Supply Card 6-3 if IN3 ≥ 1.
10-12	I3	IN4	Input control for Card 6-4 (source variable importance). Omit Card 6-4 if IN4 ≤ 0. Supply Card 6-4 if IN4 ≥ 1.
13-15	I3	IN5	Input control for Card 6-5 (group importance). Omit Card 6-5 if IN5 ≤ 0. Supply Card 6-5 if IN5 ≥ 1.
16-18	I3	IN6	Input control for Card 6-6 (linear buildup). Omit Card 6-6 if IN6 ≤ 0. Supply Card 6-6 if IN6 ≥ 1.

Column	Format	Symbol	Definition
19-21	I3	IN7	Input control for Card 6-7 (heavy element scatter). Omit Card 6-7 if IN7 $\leq$ 0. Supply Card 6-7 if IN7 $\geq$ 1.
22-24	I3	IN8	Input control for Card 6-8 (hydrogen scatter). Omit Card 6-8 if IN8 $\leq$ 0. Supply Card 6-7 if IN7 $\geq$ 1.
25-27	I3	IN9	Input control for Card 6-8 (scaling factors) Omit Card 6-9 if IN9 $\leq$ 0. Supply Card 6-9 if IN9 $\geq$ 1.
28-72	15I3	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 6-1, Descriptive Information for Section 6 Data

NOTE: a) Omit this card if IN1 $\leq$ 0.
b) Supply IN1 physical cards if IN1 $\geq$ 1.

Column	Format	Symbol	Definition
1-72	18A4	---	Any desired information for describing Section 6 data.
73-80	2A4	---	Any desired information for card identification.

CARD 6-2, Spherical Pseudo-Source and Detector

NOTE: a) Omit this card if IN2 $\leq$ 0.
b) Supply this card if IN2 $\geq$ 1.

Column	Format	Symbol	Definition
1-9	E9 0	RADIUS	Radius of a pseudo spherical source which encloses all the fixed sources (cm).
10-18	E9.0	XCT(1)	x - coordinate of the center of the sphere (cm).
19-27	E9.0	XCT(2)	y - coordinate of the center of the sphere (cm).
28-36	E9.0	XCT(3)	z - coordinate of the center of the sphere (cm).
37-45	E9.0	DELTA	Radius of a pseudo spherical detector which covers the area in space where fluxes are being calculated; the center of this sphere is the "preferred point" for surface and volume flux calculations (not used if NPOINT = 1).

Column	Format	Symbol	Definition
46-54	E9.0	BDC(1)	x - coordinate of the center of the detector sphere (cm).
55-63	E9.0	BDC(2)	y - coordinate of the center of the detector sphere (cm).
64-72	E9.0	BDC(3)	z - coordinate of the center of the detector sphere (cm).
73-80	2A4	---	Any desired information for card identification.

CARD 6-3, Relative Source Importances

NOTE: a) Omit this card if IN3 $\leq$ 0.
b) Supply this card if IN3 $\geq$ 1.
c) The data on this card is required if, and only if, MODELP = 0 and more than one source is present.

Column	Format	Symbol	Definition
1-72	8E9.0	RSI(1)	Relative importance of fixed source number 1, (use intuitive knowledge or, better yet, a point kernel calculation of fractional contributions from each source).
		⋮	
		RSI(NVMAX)	Relative importance of the last fixed source (these importances are normalized in the program).
73-80	2A4	---	Any desired information for card identification.

CARD 6-4, Source Variable Sampling (preferred values)

NOTE: a) Omit this card if IN4 $\leq$ 0.
b) Supply this card and Card 6-4' for all sources if IN4 $\geq$ 1.
c) The first three pieces of data on this card are required if MODELP = 0.
d) The last two pieces of data are required if MODELQ = 0.

Column	Format	Symbol	Definition
1-9	E9.0	VMD(1, I)	Preferred value of the first source variable of the Ith source, must be in the range of the variable including the minimum and maximum values.

Column	Format	Symbol	Definition
46-72	3E9.0	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 6-4', Source Variable Sampling (relative importance)

NOTE: a) Omit this card if IN4 ≤ 0.
b) Supply this card immediately behind Card 6-4 for the same source if IN4 ≥ 1, i.e.,

Card 6-4 }
Card 6-4' } Source 1
⋮
Card 6-4 }
Card 6-4' } Last Source

c) All numbers on this card must be greater than 0.0.

Column	Format	Symbol	Definition
1-9	E9.0	ALP(1, I)	Relative importance of the preferred value of the first source variable for the Ith source, expressed as a ratio to the importance of the value of the variable furthest away (either the minimum or maximum value of the variable). > 1.0, the preferred point is more important. 0.0 < ALP(1, I) < 1.0 the preferred point is less important. = 1.0, all points are equally important (this number must be > 0.0 since its logarithm is computed).
		⋮	
37-45	E9.0	ALP(5, I)	Relative importance of the preferred value of the fifth source variable of the Ith source.
46-72	3E9.0	---	These columns are not used and should be left blank.
73-80	2A4	---	Any desired information for card identification.

CARD 6-5, Group Importance

NOTE: a) Omit this card if IN5 ≤ 0.
b) Supply this card if IN5 ≥ 1.

Column	Format	Symbol	Definition
1-72	8E9.0	GIM(1)	Relative importance of particles in the first source and cross section group; e.g., an average flux-to-dose conversion factor for the first group.
		⋮	
		GIM(NEMAX)	Relative importance of particles in the last source and cross section group; e.g., an average flux-to-dose conversion factor for the last group.
73-80	2A4	---	Any desired information for card identification.

CARD 6-6, Linear Building Coefficients

NOTE: a) Omit this card if IN6 ≤ 0.
b) Supply this card if IN6 ≥ 1.

Column	Format	Symbol	Definition
1-72	8E9.0	AIM(1)	Linear buildup coefficient for group 1 used to estimate the importance of future collisions, this number, when multiplied by the mean free paths to a detector, is used to approximate the future scattered contributions.
		⋮	
		AIM(NEMAX)	Linear buildup coefficient for the last source and cross section group.
73-80	2A4	---	Any desired information for card identification.

CARD 6-7, Heavy Element Scattering Importance

NOTE: a) Omit this card if IN7 < 0.
b) Supply this card if IN7 ≥ 0.
c) All numbers on this card must be greater than 0.0.

Column	Format	Symbol	Definition
1-72	8E9.0	ALM(1)	Ratio of forward-to-backward scattering importance for heavy elements for the first energy group. (See note below.)
		⋮	

Column	Format	Symbol	Definition
		ALM(NEMAX)	Similar ratio for the last energy group.
73-80	2A4	---	Any desired information for card identification.

NOTES: For neutrons, a ratio of 10.0 for each group has worked well; for photons, the ratio

$$\frac{\frac{d\Sigma}{d\Omega}\text{ (0° scatter) x energy after scatter (0°)}}{\frac{d\Sigma}{d\Omega}\text{ (180° scatter) x energy after scatter (180°)}}$$

using the Klein-Nishina formula for an average group energy has yielded good results.

CARD 6-8, Hydrogen Scattering Importance

NOTE: a) Omit this card if IN8 $\leq$ 0.
b) Supply this card if IN8 $\geq$ 1.
c) All numbers on this card must be greater than 0.0.

Column	Format	Symbol	Definition
1-72	8E9.0	ALH(1)	Ratio of forward-to-backward scattering importance for hydrogen for the first energy group. (See notes below.)
		⋮	
		ALH(NEMAX)	Ratio of forward-to-backward scattering importance for hydrogen for the last energy group.
73-80	2A4	---	Any desired information for card identification.

NOTES: For neutrons, there is no back scattering from hydrogen; large ratios, e.g., 10^3 for each group have worked well. For photons, numbers identical to those on Card 6-7 have been used.

These numbers are applied only to the hydrogen density specified for the region, they are not applied to the hydrogen part, if any, of material compositions. Therefore, it is essential in neutron problems, that the hydrogen densities be specified by region to properly approximate the angular dependence of neutron scattering from hydrogen.

CARD 6-9, Sampling Parameter Scaling Factors

NOTE: a) Omit this card if IN9 $\leq$ 0.
b) Supply this card if IN9 $\geq$ 1.

Column	Format	Symbol	Definition
1-9	E9.0	ATA	Spherical pseudo source sampling, polar angle importance adjustment. 1.0, all angles equally important, >1.0, shifts importance towards small angles, <1.0, shifts importance towards large angles, Numbers in the range 1.0 $\leq$ ATA $\leq$ 10.0 work fine. (This number must be greater than 0.0.)
10-18	E9.0	ATB	Spherical pseudo source sampling, azimuthal angle important adjustment, 1.0, all angles equally important, >1.0, shifts importance towards 0°, <1.0, shifts importance away from 0°. This angle is measured in a rotated coordinate system and a little difficult to relate to the true coordinate system. The usual procedure is to use ATB = 1.0 (this number must be greater than 0.0).
19-27	E9.0	ATC	Spherical pseudo-source sampling, spatial importance adjustment. 1.0, uses built-in estimate of spatial importance >1.0, shifts importance to lower source energies (source points closer to the detector). <1.0, shifts importance to higher source energies (source points further away). General use of numbers 0.7 <ATC<1.3 yield good results. (The program can be tricked for leakage-type surface and volume detector calculations by putting the preferred detector in the center of the source and using ATC $\approx$ -1.0.)
28-36	E9.0	ATD	Flux contribution importance used in cutoff considerations; if all contributions to all detector groups on 2 successive inner iterations of a given outer iteration are less than ATD times the flux already obtained in this outer iteration, then the inner iterations are terminated.

Column	Format	Symbol	Definition
37-45	E9.0	AT	Scaling factor for the spatial importance on the first leg of the scattering triangle 1.0 uses built-in parameters. <1.0, shifts importance to higher energies >1.0, shifts importance to lower energies General use of numbers 0.6 < AT < 1.2 yields good results (must be greater than 0.0).
46-54	E9.0	BT	Scaling factor for the spatial importance on the second leg of the scattering triangle; should approximate higher order scattering effects, so it is generally less than AT. If MODELU = 0, it must be less than AT in absolute magnitude, i.e., |BT| < AT. Numbers on the order of 0.4 < BT < 0.9 yield good results. (If the trick mentioned in discussing ATC is used, it should also be used here; i.e., -0.9 < BT < -0.4.) This number cannot = 0.0.
55-63	E9.0	AS	Scaling factor for preferred direction (towards detector) importance 1.0 uses built-in parameter >1.0 forces even more <1.0 forces less. 0.0 yields no effect. (<0.0 forces away and should be used when ATC and BT are <0.0.) Use of 1.0 yields good results for point detectors.
64-72	E9.0	BS	Scaling factor for scattered direction importance 1.0, uses group averaged parameter. >1.0 forces even more <1.0, forces less. 0.0, no effect from scattered direction. Use of 1.0 yields good results.
73-80	2A4	---	Any desired information for card identification.

SECTION

10.0 SAMPLE PROBLEM

Numerical results obtained from the FASTER program for typical nuclear reactors are reported in References 17 and 18. The input data used in calculations for a large NERVA-type nuclear reactor are reproduced in the classified appendix, Reference 19.

Problem Description

A sample problem of moderate complexity is discussed below. Included in this discussion is a description of the printout obtained from running the FASTER program.

The sample problem involves the configuration used in a Lockheed study reported in Reference 20. Data taken directly from this study include the geometrical model--shown in Figure 11, material compositions; spatial source distributions; and the differential photon spectrum. The specific problem is a dose rate calculation at a point detector above a partially empty liquid hydrogen propellant tank. The problem was run for both photons and neutrons in a single computer run using the "change case" capability of the FASTER program. This problem was run for 400 outer iterations for both photons and neutrons and required less than 4 minutes on the CDC 6600 computer. Approximately twice as much time would have been required on the IBM 7094 computer.

Input Data

The complete printout for this problem is shown in Table 2. The data cards for the problems are not shown since they appear on the printout in almost the same form. The major difference is that the card identification from columns 73-80 appears on the left side of the printout. The card identification (if any) is normally followed by 3 periods (...) after which the data from columns 1 - 72 is printed. However, the printout for this problem has had the card number entered above the 3 periods to simplify the examination of data on particular cards.

The listing of the data cards for each section is preceded by a line indicating the appropriate input section. Section 1 data are first and the printout of input data continues

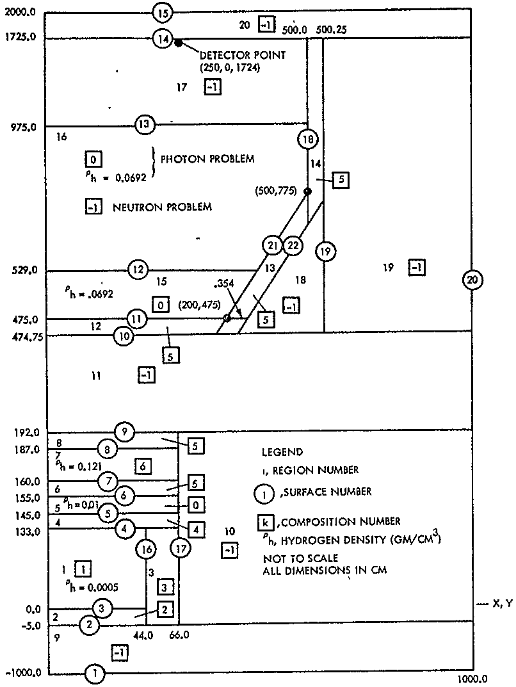


Figure 11. Sample Problem Geometry

611855-75B

Astronuclear Laboratory

TABLE 2

THE SAMPLE PROBLEM PRINTOUT IS PRESENTED ON PAGES 154 THROUGH 169

```
(F)ORTRAN,(A)NALYTIC (S)OLUTION,(T)RANSPORT (E)QUATION,(R)ANDOM SAMPLING*****THE FASTER CODE******CASE      1
MATHEMATICAL AND NUMERICAL ANALYSIS,COMPUTER PROGRAMMING AND CHECKOUT BY*****T.M.JORDAN,WANL******PAGE      1

*********************************DATA INPUT SECTION 1.  VARIABLE DIMENSIONS*********************************
INPUTS        0   1   1   1   1   1   1   1   1   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
TITLEA   ........................SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY
TITLEB   ........................EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC)
LIMITG       22   6  20   5  30
LIMITS       23   1  12  50
OPTIONX       0   1   0   6   6
LIMITX        0   0   0   0
LIMITD       11   1   0   8   1   1   1  13
              1   1   1   1   3
              1 400   0  10
*********************************DATA INPUT SECTION 2. SURFACES AND REGIONS*********************************
INPUTS        0  22  20   1   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
SURFACE       1  -0   3 -1.0000E+03 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       2  -0   3 -5.0000E+00 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       3  -0   3  0.         -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       4  -0   3  1.3300E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       5  -0   3  1.4500E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       6  -0   3  1.5500E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       7  -0   3  1.6000E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       8  -0   3  1.8700E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE       9  -0   3  1.9200E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE      10  -0   3  4.7475E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE      11  -0   3  4.7500E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE      12  -0   3  5.2900E+02 -0.         -0.         -0.         C           -0.         -0.
SURFACE      13  -0   3  9.7500E+02 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE      14  -0   3  1.7250E+03 -0.         -0.         -0.         -0.         -0.         0.
SURFACE      15  -0   3  2.0000E+03 -0.         -0.         -0.         -0.         -0.         -0.
SURFACE      16  -0  12  0.                     4.4000E+01  0.          4.4000E+01 -0.         -0.         -0.
SURFACE      17  -0  12 -0.                     6.6000E+01 -0.          6.6000E+01 -0.         -0.         -0.
SURFACE      18  -0  12 -0.                     5.0000E+02 -0.          5.0000E+02 -0.         -0.         -0.
SURFACE      19  -0  12 -0.                     5.0025E+02 -0.          5.0025E+02 -0.         -0.         -0.
SURFACE      20  -0  12 -0.                     1.0000E+03 -0.          1.0000E+03 -0.         -0.         -0.
SURFACE      21  -0   9 -0.         -0.         2.0000E+02  4.7500E+02  5.0000E+02  7.7500E+02 -0.
SURFACE      22  -0   9 -0.         -0.         2.0035E+02  4.7500E+02  5.0035E+02  7.75..E+02 -0.
REGION        1   1   1   3   4  16  -0  -0  -0  -0  -0  -0  5.0000E-04 -0.         -0.         6.0000E+01
REGION        2  -0   2   2   3  16  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.        -2.0000E+00
REGION        3  -0   3   2   4  16  17  -0  -0  -0  -0  -0 -0.         -0.         5.0000E+01  6.0000E+01
REGION        4  -0   4   4   5  17  -0  -0  -0  -0  -0  -0  4.0000E-03 -0.         -0.         1.4000E+02
REGION        5  -0   0   5   6  17  -0  -0  -0  -0  -0  -0  1.0000E-02 -0.         -0.         1.5000E+02
```



```
SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY     *****THE FASTER CODE******CASE      1
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE      2

REGION        6  -0   5   6   7  17  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.         1.5800E+02
REGION        7  -0   6   7   8  17  -0  -0  -0  -0  -0  -0  1.2100E-01 -0.         -0.         1.7000E+02
REGION        8  -0   5   8   9  17  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.         1.9000E+02
REGION        9  -0  -1   1   2  20  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.        -5.0000E+02
REGION       10  -0  -1  17  20   2   9  -0  -0  -0  -0  -0 -0.         -0.         7.0000E+01  6.0000E+01
REGION       11  -0  -1   9  10  20  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.         3.0000E+02
REGION       12  -0   5  10  11  21  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.         4.7490E+02
REGION       13  -0   5  10  18  21  22  -0  -0  -0  -0  -0 -0.         -0.         2.0010E+02  4.7500E+02
REGION       14  -0   5  14  18  19  22  10  -0  -0  -0  -0 -0.         -0.         5.0010E+02  1.0000E+03
REGION       15  -0   0  11  12  21  -0  -0  -0  -0  -0  -0  6.9200E-02 -0.         -0.         5.0000E+02
REGION       16  -0   0  12  13  18  21  -0  -0  -0  -0  -0  6.9200E-02 -0.         -0.         8.0000E+02
REGION       17  -0  -1  13  14  18  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.         1.2000E+03
REGION       18  -0  -1  10  19  22  -0  -0  -0  -0  -0  -0 -0.         -0.         3.0000E+02  4.7500E+02
REGION       19  -0  -1  19  20  10  14  -0  -0  -0  -0  -0 -0.         -0.         6.0000E+02  5.0000E+02
REGION       20  -0  -1  14  15  20  -0  -0  -0  -0  -0  -0 -0.         -0.         -0.         1.8000E+03
*********************************DATA INPUT SECTION 3.  INDEPENDENT SOURCES*********************************
INPUTS        0   1   1   0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
LEVELS       1.0000E+01  9.0000E+00  8.0000E+00  7.0000E+00  6.0000E+00  5.0000E+00  4.0000E+00  3.0000E+00
LEVELS       2.5000E+00  2.0000E+00  1.7500E+00  1.5000E+00  1.2500E+00  1.0000E+00  8.0000E-01  6.0000E-01
LEVELS       4.0000E-01  3.0000E-01  2.0000E-01  1.0000E-01  8.0000E-02  6.0000E-02  4.0000E-02  3.0000E-02
SOURCE        1   1 -12   2  12   2   2  15   2   0   0  -0  3.1200E+19 -0.         -0.         -0.
R,F(R)       0.          1.8000E+00  3.0000E+01  1.8000E+00  3.0000E+01  1.4400E+00  3.6000E+01  1.4400E+00
R,F(R)       3.6000E+01  1.3300E+00  3.9000E+01  1.3300E+00  3.9000E+01  1.3700E+00  4.2000E+01  1.3700E+00
R,F(R)       4.2000E+01  1.4900E+00  4.3000E+01  1.4900E+00  4.3000E+01  1.6600E+00  4.4000E+01  1.6600E+00
AZIMUTHL    -3.1416E+00  1.0000E+00  3.1416E+00  1.0000E+00
Z,F(Z)       0.          4.4200E-01  5.0000E+00  4.4200E-01  5.0000E+00  7.6700E-01  2.5000E+01  7.6700E-01
Z,F(Z)       2.5000E+01  1.3300E+00  6.5000E+01  1.3300E+00  6.5000E+01  1.2700E+00  1.1500E+02  1.2700E+00
Z,F(Z)       1.1500E+02  7.6000E-01  1.3100E+02  7.6000E-01  1.3100E+02  6.3200E-01  1.3300E+02  6.3200E-01
AZIMUTHL    -3.1416E+00  1.0000E+00  3.1416E+00  1.0000E+00
P-ANGLF     -1.0000E+00  1.0000E+00  1.0000E+00  1.0000E+00
SPECTRUM     8.0000E+00  4.1000E-03  7.0000E+00  9.0500E-03  6.0000E+00  2.3000E-02  5.0000E+00  6.8800E-02
SPECTRUM     4.0000E+00  2.0200E-01  3.5000E+00  3.4900E-01  3.0000E+00  6.0500E-01  2.5000E+00  1.0500E+00
SPECTRUM     2.0000E+00  1.8200E+00  1.5000E+00  3.1500E+00  1.0000E+00  5.4600E+00  7.0000E-01  7.5900E+00
SPECTRUM     5.0000E-01  9.4600E+00  3.0000E-01  1.1800E+01  1.0000E-01  1.4700E+01
*********************************DATA INPUT SECTION 4. BASIC CROSS SECTIONS*********************************
INPUTS        0   0   1   0   0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
RHO          1.0080E+00  1.0000E+00 -0.         -0.         -0.         -0.         -0.         -0.
H -XSECT     5.3700E-02  5.7100E-02  6.2100E-02  6.7400E-02  7.4700E-02  8.4000E-02  9.6900E-02  1.1600E-01
H -XSECT     1.2900E-01  1.4700E-01  1.5900E-01  1.7200E-01  1.8900E-01  2.1100E-01  2.3500E-01  2.6800E-01
H -XSECT     3.1700E-01  3.5400E-01  4.0700E-01  4.9300E-01  5.1700E-01  5.4600E-01  5.7800E-01  5.9700E-01
```

```
SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY     *****THE FASTER CODE******CASE    1
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    3

RHO  LI6 4:3  6.0150E+00  3.0000E+00 -0.         -0.         -0.         -0.         -0.         3.1000E-02
LI-ST    4:4  1.7700E-01  1.8800E-01  1.9800E-01  2.1400E-01  2.3300E-01  2.6100E-01  3.0300E-01  3.4700E-01
LI-ST    4:4  3.8800E-01  4.4200E-01  4.7600E-01  5.1500E-01  5.6500E-01  6.3500E-01  7.0000E-01  8.0500E-01
LI-ST    4:4  9.5200E-01  1.0650E+00  1.2200E+00  1.4800E+00  1.5500E+00  1.6400E+00  1.7430E+00  1.8350E+00
RHO  LI7 4:3  7.0160E+00  3.0000E+00 -0.         -0.         -0.         -0.         -0.         3.8800E-01
LI-ST    4:4  1.7700E-01  1.8800E-01  1.9800E-01  2.1400E-01  2.3300E-01  2.6100E-01  3.[illegible]E-01  3.4700E-01
LI-ST    4:4  3.8800E-01  4.4200E-01  4.7600E-01  5.1500E-01  5.6500E-01  6.35[illegible]E-01  7.0000E-01  8.0500E-01
LI-ST    4:4  9.5200E-01  1.0650E+00  1.2200E+00  1.4800E+00  1.5500E+00  1.6400E+00  1.7430E+00  1.8350E+00
RHO    C 4:3  1.2010E+01  6.0000E+00  1.3000E+00  1.7000E+00  1.2000E+00 -0.         -0.         -0.
C -XSECT 4:4  3.8600E-01  4.0500E-01  4.2400E-01  4.5300E-01  4.8800E-01  5.3800E-01  6.0600E-01  7.0900E-01
C -XSECT 4:4  7.8200E-01  8.8400E-01  9.5400E-01  1.0320E+00  1.0900E+00  1.2670E+00  1.4100E+00  1.6050E+00
C -XSECT 4:4  1.9000E+00  2.1200E+00  2.4400E+00  2.9800E+00  3.1400E+00  3.3700E+00  3.8400E+00  4.5800E+00
RHO   FE 4:3  5.5847E+01  2.6000E+01  1.1000E-01 -0.         -0.          2.0000E+00  7.0000E+00  1.2000E+00
FE-XSECT 4:4  2.7300E+00  2.7300E+00  2.7400E+00  2.7600E+00  2.8200E+00  2.9000E+00  3.0600E+00  3.3400E+00
FE-XSECT 4:4  3.6000E+00  3.9300E+00  4.1900E+00  4.4900E+00  4.9500E+00  5.5200E+00  6.1600E+00  7.0600E+00
FE-XSECT 4:4  8.5200E+00  9.8500E+00  1.2800E+01  3.1720E+01  5.0940E+01  1.0330E+02  3.2100E+02  7.3350E+02
RHO U235 4:3  2.3504E+02  9.2000E+01  2.0000E-01 -0.         -0.         -0.         -0.         -0.
U -XSECT 4:4  7.0210E+01  1.9560E+01  1.8940E+01  1.8480E+01  1.7970E+01  1.7630E+01  1.7390E+01  1.7580E+01
U -XSECT 4:4  1.7920E+01  1.9120E+01  1.9840E+01  2.1660E+01  2.4100E+01  2.9930E+01  3.7600E+01  5.3800E+01
U -XSECT 4:4  1.0230E+02  1.7850E+02  4.6240E+02  9.9330E+02  1.7480E+03  3.6000E+03  9.9630E+03  2.0355E+04
*******************SCATTERING CROSS SECTIONS REQUIRED      0 OF 10590 AVAILABLE LOCATIONS*******************
*******************************DATA INPUT SECTION 5, DETECTORS + RESPONSE*********************************
INPUTS  5:0   0   1   1   1   0   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0      -0  -0  -0
GROUPS  5:2   1   1   2   2   3   3   4   4   5   5   6   6   7   7   8   8   9   9  10  10  11  11  11
RESPONSE5:3   1   0      REM/HOUR     1.0000E+00  1.0700E-05  9.1200E-06  7.1800E-06  5.3300E-06  3.8100E-06
RESPONSE5:3'  2.9600E-06  2.2800E-06  1.5900E-06  8.1500E-07  3.7100E-07  1.3000E-07  2.8700E-07
DETECTOR5:4   1   0   0 -0.          -0.         -0.          1.0000E+00  2.5000E+02 -0.         1.7240E+03
        5:6   1   2   3   4   5   6   7   8  12  13  14  15  16
*******************************DATA INPUT SECTION 6,  SAMPLING PARAMETERS*********************************
        6:0   1   1   0   0   1   1   1   1   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
COMMENT6 6:1.........................SAMPLING PARAMETERS
        6:2   7.9000E+01  0.          0.          6.6500E+01 -0.         -0.         -0.         -0.
GROUP-IM6:5   1.0000E-05  9.5000E-06  8.6000E-06  7.6000E-06  6.8000E-06  5.8000E-06  4.8000E-06  4.0000E-06
GROUP-IM6:5   3.5000E-06  3.1000E-06  2.8000E-06  2.4000E-06  2.1000E-06  1.7000E-06  1.4000E-06  1.0000E-06
GROUP-IM6:5   7.0000E-07  4.8000E-07  2.6000E-07  1.3000E-07  1.0000E-07  1.0000E-07  1.0000E-07
        6:6   2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01
        6:6   2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01
        6:6   2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01  2.0000E-01
SCAT-RAT6:7   4.0000E+03  2.5000E+03  1.8000E+03  1.4000E+03  1.0000E+03  7.0000E+02  4.5000E+02  3.0000E+02
SCAT-RAT6:7   2.0000E+02  1.5000E+02  1.1200E+02  8.2000E+01  6.0000E+01  4.0000E+01  2.7000E+01  1.7000E+01
```


Astronuclear Laboratory

```
SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY     *****THE FASTER CODE******CASE    1
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    4

SCAT-RAT6:7   1.0000E+01  6.0000E+00  4.0000E+00  2.7000E+00  2.5000E+00  2.0000E+00  1.7000E+00
SCAT-RAT6:8   4.0000E+03  2.5000E+03  1.8000E+03  1.4000E+03  1.0000E+03  7.0000E+02  4.5000E+02  3.0000E+02
SCAT-RAT6:8   2.0000E+02  1.5000E+02  1.1200E+02  8.2000E+01  6.0000E+01  4.0000E+01  2.7000E+01  1.7000E+01
SCAT-RAT6:8   1.0000E+01  6.0000E+00  4.0000E+00  2.7000E+00  2.5000E+00  2.0000E+00  1.7000E+00
RATIOS  6:9   1.0000E+00  1.0000E+00  5.0000E-01  1.0000E-03  7.0000E-01  5.0000E-01  0.          1.0000E+00
*************************************DATA INPUT AND PREPARATION COMPLETED*********************************
*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*GO*
```


Astronuclear Laboratory

SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY *****THE FASTER CODE******CASE 1
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE 5

*********************************FLUXES FOR DETECTOR 1 AFTER 400 PACKETS***********************************

	CALCULATED PRECISION	AVERAGE ENERGY-MEV	NUMBER FLUX THIS GROUP	ENERGY FLUX THIS GROUP	NUMBER FLUX DERIVATIVE	ENERGY FLUX DERIVATIVE	NUMBER FLUX CUMULATIVE	ENERGY FLUX CUMULATIVE
GROUP 1...	0.	0.	0.	0.	0.	0.	0.	0.
GROUP 2...	7.0397E-02	6.7364E+00	1.2661E+07	8.5291E+07	6.3307E+06	4.2646E+07	1.2661E+07	8.5291E+07
GROUP 3...	1.2421E-01	4.7045E+00	6.4996E+07	3.0578E+08	3.2498E+07	1.5289E+08	7.7657E+07	3.9107E+08
GROUP 4...	1.5024E-01	3.1211E+00	1.2743E+08	3.9772E+08	8.4953E+07	2.6515E+08	2.0509E+08	7.8879E+08
GROUP 5...	2.6825E-01	2.0679E+00	1.5960E+08	3.3003E+08	2.1280E+08	4.4004E+08	3.6469E+08	1.1188E+09
GROUP 6...	2.4503E-01	1.4820E+00	1.4655E+08	2.1718E+08	2.9309E+08	4.3435E+08	5.1123E+08	1.3360E+09
GROUP 7...	2.6005E-01	9.6454E-01	2.8193E+08	2.7194E+08	6.2652E+08	6.0430E+08	7.9317E+08	1.6079E+09
GROUP 8...	3.1338E-01	5.5938E-01	5.5496E+08	3.1043E+08	1.3874E+09	7.7608E+08	1.3481E+09	1.9184E+09
GROUP 9...	3.3068E-01	2.8289E-01	1.6013E+08	4.5300E+07	8.0065E+08	2.2650E+08	1.5083E+09	1.9637E+09
GROUP 10...	6.0923E-01	1.1708E-01	1.1227E+09	1.3145E+08	9.3555E+09	1.0954E+09	2.6309E+09	2.0951E+09
GROUP 11...	6.0018E-01	6.3239E-02	7.5916E+07	4.8008E+06	1.5183E+09	9.6017E+07	2.7068E+09	2.0999E+09

*************************NUMBER FLUX RESPONSES FOR DETECTOR 1 AFTER 400 PACKETS*************************

	REM/HOUR
GROUP 1...	0.
GROUP 2...	9.9952E+01
GROUP 3...	3.8879E+02
GROUP 4...	5.6571E+02
GROUP 5...	5.2991E+02
GROUP 6...	3.8036E+02
GROUP 7...	5.1941E+02
GROUP 8...	6.2366E+02
GROUP 9...	8.8876E+01
GROUP 10...	2.2956E+02
GROUP 11...	1.3865E+01
TOTALS.......	3.4401E+03
MIN ERROR...	9.8763E-02
MAX ERROR...	2.5690E-01

*************************NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS*************************

	ITERANT 1	REGION 1	REGION 2	REGION 3	REGION 4	REGION 5	REGION 6	REGION 7
GROUP 1...	0.	0.	0.	0.	0.	0.	0.	0.
GROUP 2...	1.2661E+07	8.0901E+04	0.	7.0586E+03	0.	0.	1.0765E+05	4.2188E+05
GROUP 3...	6.4996E+07	1.0890E+06	0.	1.3236E+05	3.8269E+05	0.	2.2475E+06	1.0407E+07
GROUP 4...	1.2743E+08	3.1946E+06	0.	1.3947E+06	1.4615E+06	0.	8.9274E+06	2.1597E+07
GROUP 5...	1.5960E+08	1.8013E+06	0.	4.8757E+06	1.7619E+06	0.	4.6404E+06	1.7124E+07
GROUP 6...	1.4655E+08	7.4252E+05	0.	5.2407E+06	1.3449E+06	0.	3.0568E+06	6.0527E+06
GROUP 7...	2.8193E+08	1.7002E+05	0.	6.3852E+06	3.4080E+05	0.	1.5097E+06	1.0528E+06
GROUP 8...	5.5496E+08	8.2677E+03	0.	3.6957E+06	2.5689E+06	0.	8.2937E+04	5.0908E+05

SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY *****THE FASTER CODE******CASE 1
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE 6

GROUP 9...	1.6013E+08	4.3394E+01	0.	2.5557E+05	1.4762E+05	0.	2.6323E+03	5.0310E+03
GROUP 10...	1.1227E+09	8.8622E-05	3.0088E-55	8.8383E+03	7.1608E+03	0.	3.2657E-01	1.8086E+00
GROUP 11...	7.5916E+07	6.9577E-33	9.0566-150	5.0195E+03	7.9767E-05	0.	4.1301E-11	8.0481E-11
REM/HOUR	3.4401E+03	2.9566E+01	6.1523E-62	5.2891E+01	2.1716E+01	0.	8.0144E+01	2.3654E+02

*************************NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS*************************

	REGION 8	REGION 12	REGION 13	REGION 14	REGION 15	REGION 16	SCATTER 0	SCATTER 1
GROUP 1...	0.	0.	0.	0.	0.	0.	0.	0.
GROUP 2...	0.	0.	0.	0.	2.1977E+05	6.6297E+05	1.1161E+07	1.5002E+06
GROUP 3...	1.4729E+06	0.	0.	0.	4.1814E+06	7.4966E+06	3.7586E+07	2.3676E+07
GROUP 4...	9.3473E+06	0.	0.	0.	6.1216E+06	3.5461E+07	3.9924E+07	6.6118E+07
GROUP 5...	7.0088E+06	0.	1.4469E+05	0.	2.6314E+06	1.0826E+08	1.1348E+07	7.9286E+07
GROUP 6...	6.5208E+06	0.	6.7360E+06	0.	2.1508E+06	1.1234E+08	2.3628E+06	6.2395E+07
GROUP 7...	3.0546E+06	9.1398E+03	3.9501E+07	6.3105E+07	4.9943E+06	1.6150E+08	3.1205E+05	8.6505E+07
GROUP 8...	4.5719E+05	8.5341E+02	1.6128E+07	8.8349E+07	4.2717E+05	4.4273E+08	9.5295E+03	8.7282E+07
GROUP 9...	1.5765E+04	1.6136E+03	1.3250E+07	3.3616E+07	3.0219E+04	1.1281E+08	1.2035E+01	9.9542E+05
GROUP 10...	6.1644E+02	3.9678E+00	2.6209E+06	4.4848E+07	4.5514E+03	1.0752E+09	2.2818E-09	7.8979E+02
GROUP 11...	9.1599E-02	2.1657E-02	5.7032E+01	1.4080E+06	3.7175E-01	7.4503E+07	0.	0.
REM/HOUR	9.6652E+01	1.8694F-02	1.1675E+02	2.4363E+02	7.7941E+01	1.9497E+03	5.3457E+02	1.1302E+03

*************************NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS*************************

	SCATTER 2	SCATTER 3	SCATTER 4	SCATTER 5	SCATTER 6	SCATTER 7	ANGULAR 1	SPATIAL 1
GROUP 1...	0.	0.	0.	0.	0.	0.	0.	0.
GROUP 2...	0.	0.	0.	0.	0.	0.	1.2504E+07	1.2367E+07
GROUP 3...	3.7332E+06	-0.	0.	0.	0.	0.	6.4155E+07	6.3482E+07
GROUP 4...	2.1388E+07	0.	0.	0.	0.	0.	1.2545E+08	1.2440E+08
GROUP 5...	5.6404E+07	1.2561E+07	0.	0.	0.	0.	1.5761E+08	1.6027E+08
GROUP 6...	3.8345E+07	3.2233E+07	1.1213E+07	0.	0.	0.	1.4306E+08	1.4781E+08
GROUP 7...	1.2786E+08	3.3986E+07	3.2193E+07	1.0838E+06	0.	0.	2.5806E+08	3.0563E+08
GROUP 8...	1.3070E+08	1.9135E+08	8.7381E+07	5.8241E+07	0.	0.	5.0989E+08	6.0829E+08
GROUP 9...	8.0511E+06	2.9793E+07	6.0951E+07	5.1079E+07	9.1694E+06	8.9065E+04	1.4082E+08	1.7970E+08
GROUP 10...	1.9769E+07	2.4815E+07	2.6005E+07	3.4423E+07	1.0725E+08	1.4140E+08	9.7146E+08	1.3502E+09
GROUP 11...	6.9574E-33	6.4315E+02	1.6224E+04	2.1545E+05	6.9278E+05	2.0099E+07	7.2435E+07	9.6058E+07
REM/HOUR	7.9501E+02	4.2462E+02	2.2576E+02	1.0287E+02	2.7109E+01	3.2634E+01	3.2725E+03	3.5854E+03

```
SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY     *****THE FASTER CODE******CASE    1
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    7

***********************BOUNDARY SEARCH PARAMETERS. (SURFACE,MOST PROBABLE NEXT REGION)**********************
REGION    1  (  -3,  2)  (   4,  4)  (  16,  3)
REGION    2  (  -2,  9)  (   3,  1)  (  16,  3)
REGION    3  (  -2,  9)  (   4,  4)  ( -16,  1)  (  17, 10)
REGION    4  (  -4,  1)  (   5,  5)  (  17, 10)
REGION    5  (  -5,  4)  (   6,  6)  (  17, 10)
REGION    6  (  -6,  5)  (   7,  7)  (  17, 10)
REGION    7  (  -7,  6)  (   8,  8)  (  17, 10)
REGION    8  (  -8,  7)  (   9, 11)  (  17, 10)
REGION    9  (  -1, 31)  (   2,  0)  (  20, 31)
REGION   10  ( -17,  6)  (  20, 31)  (  -2,  9)  (   9, 11)
REGION   11  (  -9,  8)  (  10, 18)  (  20, 31)
REGION   12  ( -10, 11)  (  11, 15)  ( -21,  0)
REGION   13  ( -10,  0)  (  18,  0)  (  21, 16)  ( -22, 18)
REGION   14  (  14,  0)  ( -18, 17)  (  19, 19)  ( -22,  0)  ( -10,  0)
REGION   15  ( -11, 12)  (  12, 16)  ( -21, 13)
REGION   16  ( -12, 15)  (  13, 17)  (  18, 14)  ( -21, 13)
REGION   17  ( -13, 16)  (  14, 20)  (  18, 14)
REGION   18  ( -10, 11)  (  19, 19)  (  22, 13)
REGION   19  ( -19,  0)  (  20, 31)  ( -10, 11)  (  14, 20)
REGION   20  ( -14,  0)  (  15, 31)  (  20, 31)
```



```
SAMPLE PROBLEM 1. PHOTON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY     *****THE FASTER CODE******CASE    2
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    1

********************************DATA INPUT SECTION 1.  VARIABLE DIMENSIONS*********************************
INPUTS 1/-0   0   1   0   0   1   1   1   1   0   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
TITLEA  1-2.........................SAMPLE PROBLEM 2. NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY
LIMITS  1-5  13   1  12  50
OPTIONX 1-6   1   1   0   6   6
        1-7   1   4  13  13
LIMITO  1-8  13   3   0   9   1   1   1  12
        1-10  1 400   0  20
********************************DATA INPUT SECTION 2. SURFACES AND REGIONS*********************************
        2-0   0   0   1   1   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
REGION  2-3  16   0  -1  12  13  18  21  -0  -0  -0  -0  -0 -0.          -0.          -0.          8.0000E+02
********************************DATA INPUT SECTION 3.  INDEPENDENT SOURCES**********************************
INPUTS 33-0   0   1   1   0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
LEVELS  3-2  1.0000E+01  7.7880E+00  6.0650E+00  3.6790E+00  2.8650E+00  2.2310E+00  1.0540E+00  8.2100E-01
LEVELS  3-2  6.3900E-01  3.8800E-01  3.0200E-01  1.8300E-01  1.1100E-01  1.0000E-02
SOURCE  3-3   1   1  -1  -1  -1  -1  -1  13   0   2   2  -0  7.8100E+19 -0.         -0.         -0.
ENERGIES3-5  1.0000E+01  7.7880E+00  6.0650E+00  3.6790E+00  2.8650E+00  2.2310E+00  1.0540E+00  8.2080E-01
ENERGIES3-5  6.3930E-01  3.8770E-01  3.0200E-01  1.8320E-01  1.1110E-01  1.0000E-02
SPECTRUM3-5  5.9640E-03  1.7060E-02  1.0820E-01  9.6320E-02  1.1410E-01  3.3010E-01  8.1610E-02  6.4950E-02
SPECTRUM3-5  8.7240E-02  2.7430E-02  3.4040E-02  1.7140E-02  1.5630E-02
********************************DATA INPUT SECTION 4. BASIC CROSS SECTIONS*********************************
INPUTS 44-0   0   1   1   0   0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
AVERAGE 4-2  8.8940E+00  6.9270E+00  4.8720E+00  3.2720E+00  2.5480E+00  1.6420E+00  9.3700E-01  7.3000E-01
AVERAGE 4-2  5.1300E-01  3.4500E-01  2.4200E-01  1.4700E-01  6.0000E-02
RHO    H4-3  1.0080E+00  1.0000E+00 -0.         -0.         -0.         -0.         -0.         -0.
HYDROGEN4-4  1.0250E+00  1.2620E+00  1.7260E+00  2.1650E+00  2.5350E+00  3.4430E+00  4.4350E+00  5.0650E+00
HYDROGEN4-4  6.1260E+00  7.3750E+00  8.7310E+00  1.1017E+01  1.7053E+01
RHO' LI64-3  6.0150E+00  3.0000E+00 -0.         -0.         -0.         -0.         -0.          3.1000E-02
LI6-ST  4-4  4.5590E-01  6.7620E-01  1.2320E+00  1.5380E+00  1.5740E+00  1.4520E+00  1.4660E+00  1.5150E+00
LI6-ST  4-4  1.8520E+00  4.2400E+00  4.9180E+00  1.6860E+00  3.2960E+00
LI6-LDI 4-5  -2   4   0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
LI6-P0D04-6  8.3890E-01  9.2140E-01  1.2940E+00  8.2750E-01  5.9620E-01  8.9040E-01  3.0890E-01  2.5790E-01
LI6-P0D04-6  4.3500E-01  2.4610E-01  1.9820E+00  6.4880E-01  1.0040E+00
LI6-P0D1.|.  3.3360E-01  6.2250E-01  5.7120E-01  7.1340E-01  1.1940E+00  3.5040E-01  5.0900E-01  8.7440E-01
LI6-P0D1.|.  4.2230E-01  1.7100E+00  1.7060E+00  4.4930E-01
LI6-P0D2.|.  1.1090E-01  2.9000E-02  2.4190E-01  5.2540E-01  0.          1.6480E-01  4.8290E-01  9.6100E-02
LI6-P0D2.|.  3.2300E-01  3.5880E-01  2.9600E-02
LI6-P0D3.|.  0.          0.          4.6750E-02  0.          0.          2.9720E-02  0.          0.
LI6-P0D3.V.  0.          0.
LI6-P1004-6  2.2090E+00  2.3880E+00  3.0180E+00  2.0240E+00  1.4300E+00  1.0960E+00  7.0510E-01  5.8060E-01
```

```
SAMPLE PROBLEM 2. NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY   *****THE FASTER CODE******CASE     2
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WAHL******PAGF     2

LI6-P1D04-6  6.1810E-01  5.6190E-01  3.2930E+00  1.1360E+00  4.3010E-01
LI6-P1D14-6  5.4850E-01  5.4800E-01  4.2560E-01  7.6190E-01 -2.4200E-01 -1.1960E-01  4.2930E-01 -1.6830E-01
LI6-P1D1...  -2.0370E-01 -5.0670E-01 -3.1890E-01 -1.9240E-01
LI6-P1D2...  -1.4180E-01 -7.0700E-02 -2.9330E-01 -7.8220E-01  0.          -2.3230E-01 -7.2880E-01 -2.3780E-01
LI6-P1D2...  -5.3780E-01 -8.8150E-01 -7.7700E-02
LI6-P1D3...  0.          0.          -1.0790E-01  0.          0.          -6.5400E-02  0.          0.
LI6-P1D34-6  0.          0.
RHO  LI74-3  7.0160E+00  3.0000E+00 -0.          -0.          -0.          -0.          -0.          3.8800E-01
LI7-ST  4-4  4.1700E-01  6.2950E-01  1.1620E+00  1.4270E+00  1.4210E+00  1.2290E+00  1.1850E+00  1.1890E+00
LI7-ST  4-4  1.2290E+00  2.4110E+00  3.1150E+00  7.9150E-01  8.7610E-01
LI7-LDI 4-5  -2   4   0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
LI7-P0D04-6  8.6200E-01  9.5670E-01  1.3550E+00  8.8360E-01  6.4150E-01  9.5710E-01  3.4340E-01  2.8870E-01
LI7-P0D04-6  4.9650E-01  2.6720E-01  2.1850E+00  6.9000E-01  1.0040E+00
LI7-P0D1...  3.2640E-01  6.0440E-01  5.7990E-01  7.6050E-01  1.1490E+00  3.3840E-01  5.6900E-01  9.1500E-01
LI7-P0D1...  4.4210E-01  1.7640E+00  1.4900E+00  3.9840E-01
LI7-P0D2...  8.5800E-02  7.4900E-03  2.0620E-01  4.2620E-01  0.          1.3170E-01  3.9000E-01  2.5000E-02
LI7-P0D2...  2.3130E-01  1.0340E-01  6.2000E-03
LI7-P0D3...  0.          0.          1.2400E-02  0.          0.          7.5500E-03  0.          0.
LI7-P0D3...  0.          0.
LI7-P1D0...  2.2400E+00  2.4380E+00  3.0300E+00  2.0960E+00  1.4840E+00  9.8980E-01  7.4550E-01  6.1560E-01
LI7-P1D0...  5.6510E-01  5.7150E-01  3.1000E+00  1.0960E+00  3.8480E-01
LI7-P1D1...  4.6270E-01  3.9960E-01  2.8920E-01  5.7140E-01 -3.7310E-01 -1.4570E-01  2.8630E-01 -4.2900E-01
LI7-P1D1...  -3.3720E-01 -1.1200E+00 -1.3840E+00 -2.0860E-01
LI7-P1D2...  -1.3160E-01 -2.0300E-02 -3.1320E-01 -7.3810E-01  0.          -2.0850E-01 -6.8110E-01 -6.8000E-02
LI7-P1D2...  -4.0230E-01 -2.8060E-01 -1.8000E-02
LI7-P1D3...  0.          0.          -3.1600E-02  0.          0.          -1.8300E-02  0.          -0.
LI7-P1D34-6  0.          0.
RHO    C4-3  1.2010E+01  6.0000E+00  1.3000E+00  1.7000E+00  1.2000E+00 -0.          -0.          -0.
 ST   C 4-4  8.9573E-01  9.3957E-01  1.3793E+00  1.9538E+00  1.6380E+00  1.8946E+00  2.4107E+00  2.6626E+00
 ST   C 4-4  3.0348E+00  3.3893E+00  3.6713E+00  3.9804E+00  4.3525E+00
LDI-C   4-5  -2   3   3  12   5  11  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
 SS   C 4-6  5.7010E-01  4.8204E-01  9.7411E-01  8.2108E-01  6.3918E-01  1.5851E+00  1.2038E+00  1.3299E+00
 SS   C 4-6  2.3633E+00  1.5810E+00  2.8307E+00  2.8396E+00  4.6099E+00
 D1   C ...  3.1784E-01  3.9527E-01  5.1330E-01  1.0514E+00  1.0206E+00  5.2091E-01  1.4747E+00  1.7005E+00
 D1   C ...  9.7611E-01  2.1491E+00  1.1557E+00  1.4172E+00
 D2   C ...  3.9040E-02  0.          1.0868E-01  1.4851E-01  0.          3.3419E-02  8.7174E-02  0.
 D2   C ...  5.6530E-02  0.          0.
 AS0  C ...  1.3672E+00  1.0494E+00  1.4895E+00  1.8481E+00  1.1781E+00  1.1128E+00  2.0312E+00  2.1942E+00
 AS0  C ...  1.8499E+00  2.5420E+00  1.8585E+00  2.1616E+00  7.7254E-01
 AS1  C 4-6  4.9294E-02 -3.7204E-01 -4.4736E-01 -1.2881E+00 -1.1126E+00 -2.9932E-01 -7.4989E-01 -1.0989E+00
```



```
SAMPLE PROBLEM 2. NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY   *****THE FASTER CODE******CASE     2
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WAHL******PAGE     3

 AS1  C 4-6 -6.3655E-01 -1.5192E+00 -9.1333E-01 -1.3322E+00
 AS2  C 4-6 -9.9660E-02  0.          -2.7039E-01 -3.6699E-01  0.          -7.9330E-02 -2.1633E-01  0.
 AS2  C 4-6 -1.3016E-01  0.          0.
 T G1 C 4-7  0.          0.          1.2558E-01  1.5104E-01  7.7926E-02  2.8750E-03  9.3270E-04  5.4130E-04
 T G1 C 4-7  4.6720E-04  1.0750E-04  8.0900E-05  3.2340E-05
 T G2 C 4-7  0.          0.          0.          8.0320E-02  2.0483E-01
 T G3 C 4-7  0.          0.          0.          3.8970E-03  1.2870E-02  6.7110E-03  6.5470E-03  9.8150E-04
 T G3 C 4-7  1.7810E-03  8.5700E-04  2.6670E-04
RHO   FE4-3  5.5847E+01  2.6000E+01  1.1000E-01 -0.          -0.          2.0000E+00  7.0000E+00  1.2000E+00
 ST  FE 4-4  1.7609E+00  1.8465E+00  2.2369E+00  2.0968E+00  2.2055E+00  2.0395E+00  1.9117E+00  2.1432E+00
 ST  FE 4-4  3.1222E+00  2.5840E+00  2.7164E+00  3.7268E+00  6.5578E+00
LDI-FE  4-5  -2   2   7  13  12  11  10   9   8   7  -0  -0  -0  -0  -0  -0
 SS   FE4-6  1.9941E+00  2.2262E+00  2.2972E+00  1.9805E+00  1.9086E+00  1.9729E+00  2.1071E+00  2.2757E+00
 SS   FE4-6  3.3342E+00  2.5178E+00  2.7389E+00  3.6650E+00  6.7542E+00
 D1   FE...  5.3260E-02  7.2683E-02  9.5957E-02  1.5842E-01  1.9199E-01  7.7779E-02  2.7935E-01  2.8254E-01
 D1   FE...  2.8369E-01  3.5347E-01  1.4220E-01  2.1681E-01
 AS0  FE...  5.0359E+00  5.4773E+00  4.6691E+00  3.2394E+00  2.5038E+00  1.6972E+00  2.1534E+00  1.5162E+00
 AS0  FE...  1.7343E+00  1.1784E+00  6.3131E-01  6.9858E-01  6.2892E-01
 AS1  FE...  2.9889E-02  3.2910E-02 -4.9720E-02 -1.1010E-01 -1.7556E-01 -6.6810E-02 -2.0460E-01 -2.5700E-01
 AS1  FE4-6 -2.3173E-01 -2.9987E-01 -1.1980E-01 -2.1540E-01
 T G1 FE4-7  9.7803E-03  4.1974E-02  2.1761E-01  1.6833E-01  1.8104E-01  4.4147E-01  8.9481E-02  6.4125E-02
 T G1 FF4-7  7.4024E-02  1.9513E-02  2.0594E-02  8.1905E-03  5.0670E-03
 T G2 FE...  1.6918E-02  1.7455E-01  1.5576E-01  1.8152E-01  4.8606E-01  1.0593E-01  7.7610E-02  9.1646E-02
 T G2 FE...  2.4610E-02  2.6250E-02  1.0549E-02  6.5790E-03
 T G3 FE...  8.7458E-02  1.4874E-01  1.8487E-01  5.3999E-01  1.2549E-01  9.3829E-02  1.1314E-01  3.0911E-02
 T G3 FE...  3.3303E-02  1.3512E-02  8.4920E-03
 T G4 FE...  0.          2.4084E-01  3.2113E-01  5.5709E-02  4.5313E-02  1.4177E-01  4.4860E-02  7.4787E-02
 T G4 FE...  3.7963E-02  3.6492E-02
 T G5 FE...  0.          8.1455E-01  0.          8.3833E-03  2.5080E-02  7.1588E-03  6.3636E-03  7.0768E-03
 T G5 FE...  9.8782E-03
 T G6 FE...  1.1129E-01  8.8503E-02  9.7840E-02  1.2615E-01  4.0676E-02  6.3899E-02  0.          5.9590E-04
 T G7 FE4-7  0.          0.          0.          0.          6.4243E-03  9.5891E-02  6.7583E-02
RHO U2354-3  2.3504E+02  9.2000E+01  2.0000E+01 -0.          -0.          -0.          -0.          -0.
 ST   U5 4-4 2.5055E+00  3.0985E+00  4.1328E+00  4.7245E+00  5.1626E+00  5.1529E+00  5.1271E+00  5.2294E+00
 ST   U5 4-4 5.9259E+00  7.0168E+00  8.0903E+00  9.7811E+00  1.4559E+01
LDI-U5  4-5  -2   2  13  13  12  11  10   9   8   7   6   5   4   3   2   1
 SS   U54-6  2.9365E+00  3.7367E+00  5.2109E+00  4.9526E+00  4.3373E+00  3.8614E+00  3.8618E+00  4.1047E+00
 SS   U54-6  4.9039E+00  5.8900E+00  7.0164E+00  8.4298E+00  1.0221E+01
 D1   U54-6  1.3391E-02  1.3309E-02  2.3410E-02  4.7280E-02  6.2644E-02  3.2946E-02  8.8340E-02  9.5222E-02
 D1   U54-6  6.2655E-02  1.5989E-01  9.6470E-02  1.4114E-01
```

```
SAMPLE PROBLEM 2. NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY   *****THE FASTER CODE******CASE    2
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    4

 AS0  U54-6  9.8067E+00  1.1781E+01  1.7629E+01  1.5815E+01  1.2722E+01  5.3946E+00  6.379XE+00  5.6056E+00
 AS0  U54-6  5.3130E+00  5.2723E+00  4.9838E+00  4.1069E+00  1.8174E+00
 AS1  U54-6 -2.1603E+00 -1.7354E+00 -4.6092E+00 -5.0163E+00 -5.0660E+00 -4.5127E-01 -2.3266E+00 -1.2098E+00
 AS1  U54-6 -1.1314E+00 -1.2249E+00 -1.0806E+00 -6.9858E-01
 T 81 U54-7  1.5491E-05  2.1290E-04  6.0079E-03  1.1821E-02  2.1765E-02  1.0805E-01  3.5893E-02  2.9917E-02
 T 81 U54-7  4.0599E-02  1.2193E-02  1.3882E-02  5.9407E-03  3.8946E-03
 T 82 U5.|.  2.0595E-04  1.1980E-02  2.9597E-02  6.2942E-02  3.0382E-01  1.4313E-01  1.2321E-01  1.7765E-01
 T 82 U5.|.  5.5267E-02  6.4250E-02  2.8050E-02  1.8684E-02
 T 83 U5.|.  5.3868E-03  2.3332E-02  6.4144E-02  5.7627E-01  2.6983E-01  2.8665E-01  3.9787E-01  1.3272E-01
 T 83 U5.|.  1.6088E-01  7.3110E-02  5.0291E-02
 T 84 U5.|.  6.1788E-03  3.2907E-02  4.5314E-01  2.6244E-01  2.7070E-01  4.5686E-01  1.6261E-01  2.0490E-01
 T 84 U5.|.  9.6667E-02  6.8486E-02
 T 85 U5.|.  1.1725E-02  3.6751E-01  2.3837E-01  2.6367E-01  4.7937E-01  1.8053E-01  2.3558E-01  1.1477E-01
 T 85 U5.|.  8.3486E-02
 T 86 U5.|.  1.1668E-01  1.6710E-01  1.9240E-01  4.2205E-01  1.8341E-01  2.6229E-01  1.3947E-01  1.0901E-01
 T 87 U5.|.  4.3287E-02  1.2058E-01  3.3228E-01  1.6922E-01  2.6752E-01  1.5589E-01  1.3120E-01
 T 88 U5.|.  5.3007E-02  2.9781E-01  1.6736E-01  2.8246E-01  1.7477E-01  1.5489E-01
 T 89 U5.|.  1.3608E-01  1.5374E-01  2.9133E-01  2.0050E-01  1.9409E-01
 T 810U5.|.  7.2078E-02  2.9765E-01  2.3165E-01  2.4862E-01
 T 811U5.V.  1.4429E-01  2.3498E-01  2.8628E-01
 T 812U5.V.  1.9649E-01  2.5482E-01
 T 813U54-7  7.2141E-02
*********************SCATTERING CROSS SECTIONS REQUIRED   577 OF 10480 AVAILABLE LOCATIONS*******************
*********************************DATA INPUT SECTION 5. DETECTORS + RESPONSE**********************************
INPUTS 55-0   0   0   3   0   0  -0   -0  -0  -0   -0  -0  -0   -0  -0  -0   -0  -0  -0   -0  -0  -0   -0  -0  -0
RESPONSE5-3   1   0   RAD(TIS)/HR      1.0000E+00  1.8000E-05  1.8000E-05  1.6500E-05  1.6000E-05  1.2600E-05
RESPONSE5-3' 1.1300E-05  8.6000E-06  7.2000E-06  6.2000E-06  5.2000E-06  4.0000E-06  3.4000E-06  2.5000E-06
RESPONSE5-3' 0.
RESPONSE5-3   2   0   RAD(ETH)/HR      1.0000E+00  2.7000E-05  2.7000E-05  2.3000E-05  2.1500E-05  1.9000E-05
RESPONSE5-3' 1.6200E-05  1.1300E-05  1.0080E-05  9.0000E-06  7.4000E-06  5.8000E-06  5.0000E-06  3.6000E-06
RESPONSE5-3' 0.
RESPONSE6-3   3   0      REM/HOUR      1.0000E+00  1.1500E-04  1.1500E-04  1.0700E-04  1.1000E-04  9.5000E-05
RESPONSE5-3' 9.5000E-05  9.0000E-05  7.4000E-05  6.5000E-05  5.4000E-05  4.0000E-05  3.3000E-05  2.1000E-05
RESPONSE5-3' 0.
**********************************DATA INPUT SECTION 6.  SAMPLING PARAMETERS*********************************
INPUTS 66-0   0   0   0   0   1   1   1   1   1  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0  -0
GROUPIMP6-5  1.8000E-05  1.7200E-05  1.6200E-05  1.4400E-05  1.2000E-05  1.0000E-05  7.9000E-06  6.7000E-06
GROUPIMP6-5  5.7000E-06  4.6000E-06  3.7000E-06  2.9500E-06  2.0000E-06
ATTEN-IM6-6  1.0000E-01  1.0000E-01  1.0000E-01  1.0000E-01  1.0080E-01  1.0000E-01  1.0000E-01  1.0000E-01
ATTEN-IM6-6  1.0000E-01  1.0000E-01  1.0000E-01  1.0000E-01  1.0800E-01
```



```
SAMPLE PROBLEM 2. NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY   *****THE FASTER CODE******CASE    2
EMPTY PROPELLANT TANK. REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    5

SCAT-RAT6-7  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01
SCAT-RAT6-7  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01  1.0000E+01
SCAT-RAT6-8  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03
SCAT-RAT6-8  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03  1.0000E+03
        6-9  1.0000E+00  1.0000E+00  5.0000E-01  1.0000E-03  7.0000E-01  5.0000E-01  0.          5.0000E-01
*************************************DATA INPUT AND PREPARATION COMPLETED************************************
*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*G0*
```

***********************************FLUXES FOR DETECTOR 1 AFTER 400 PACKETS***********************************

	CALCULATED PRECISION	AVERAGE ENERGY-MEV	NUMBER FLUX THIS GROUP	ENERGY FLUX THIS GROUP	NUMBER FLUX DERIVATIVE.	ENERGY FLUX DERIVATIVE	NUMBER FLUX CUMULATIVE	ENERGY FLUX CUMULATIVE
GROUP 1...	1.2053E-01	8.6185E+00	2.1542E+06	1.8566E+07	9.7386E+05	8.3932E+06	2.1942E+06	1.8566E+07
GROUP 2...	4.2116E-01	6.9695E+00	9.3731E+06	6.5326E+07	5.4400E+06	3.7914E+07	1.1527E+07	8.3891E+07
GROUP 3...	3.8512E-01	4.7162E+00	4.0302E+07	1.9007E+08	1.6891E+07	7.9660E+07	5.1829E+07	2.7396E+08
GROUP 4...	3.8927E-01	3.2740E+00	2.0397E+07	6.6780E+07	2.5057E+07	8.2039E+07	7.2226E+07	3.4074E+08
GROUP 5...	5.4402E-01	2.5479E+00	3.9871E+07	1.0159E+08	6.2888E+07	1.6023E+08	1.1210E+08	4.4233E+08
GROUP 6...	6.1307E-01	1.4163E+00	3.6454E+08	5.1630E+08	3.0972E+08	4.3866E+08	4.7464E+08	9.5863E+08
GROUP 7...	5.8004E-01	9.3843E-01	3.9628E+07	3.7188E+07	1.7008E+08	1.5961E+08	5.1626E+08	9.9581E+08
GROUP 8...	6.1600E-01	7.0991E-01	7.9277E+07	5.6279E+07	4.3559E+08	3.0923E+08	5.9554E+08	1.0521E+09
GROUP 9...	6.0213E-01	5.2825E-01	8.3204E+07	4.3953E+07	3.3149E+08	1.7511E+08	6.7875E+08	1.0960E+09
GROUP 10...	8.3453E-01	3.7470E-01	8.2904E+07	3.1064E+07	9.6400E+08	3.6121E+08	7.6165E+08	1.1271E+09
GROUP 11...	5.6267E-01	2.4982E-01	2.8943E+07	7.2303E+06	2.4321E+08	6.0759E+07	7.9059E+08	1.1343E+09
GROUP 12...	6.0960E-01	1.6553E-01	1.7252E+07	2.8557E+06	2.3961E+08	3.9662E+07	8.0784E+08	1.1372E+09
GROUP 13...	9.1825E-01	6.1254E-02	8.0437E+07	4.9271E+06	7.9640E+08	4.8783E+07	8.8828E+08	1.1421E+09

************************NUMBER FLUX RESPONSES FOR DETECTOR 1 AFTER 400 PACKETS************************

	RAD(TIS)/HR	RAD(ETH)/HR	REM/HOUR
GROUP 1...	3.8775E+01	5.8163E+01	2.4773E+02
GROUP 2...	1.6204E+02	2.3526E+02	1.0423E+03
GROUP 3...	6.5358E+02	8.9276E+02	4.3806E+03
GROUP 4...	2.9388E+02	4.1316E+02	2.0914E+03
GROUP 5...	4.8044E+02	7.0171E+02	3.7877E+03
GROUP 6...	3.4380E+03	4.6691E+03	3.3370E+04
GROUP 7...	3.1328E+02	4.2225E+02	3.2520E+03
GROUP 8...	5.2240E+02	7.4438E+02	5.4310E+03
GROUP 9...	4.7915E+02	6.9010E+02	5.0044E+03
GROUP 10...	4.1572E+02	5.9298E+02	4.2973E+03
GROUP 11...	1.0816E+02	1.5771E+02	1.0689E+03
GROUP 12...	5.4889E+01	8.0399E+01	5.1908E+02
GROUP 13...	1.0205E+02	1.4695E+02	8.5719E+02
TOTALS......	7.0624E+03	9.8050E+03	6.5349E+04
MIN ERROR...	3.1490E-01	3.0951E-01	3.2971E-01
MAX ERROR...	5.8551E-01	5.8557E-01	5.9717E-01

************************NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS************************

	ITERANT 1	REGION 1	REGION 2	REGION 3	REGION 4	REGION 5	REGION 6	REGION 7
GROUP 1...	2.1542E+06	1.0701E+05	0.	6.7903E+04	3.4293E+04	8.0106E+01	3.4430E+04	3.1488E+05
GROUP 2...	9.3731E+06	2.3580E+05	0.	3.1241E+05	6.6207E+04	6.1688E+02	4.1745E+04	5.2598E+05
GROUP 3...	4.0302E+07	1.0882E+05	0.	1.6401E+06	1.4962E+05	1.6927E+03	2.7376E+04	4.4064E+05
GROUP 4...	2.0397E+07	1.5540E+04	0.	7.7371E+05	8.1115E+04	1.3636E+02	5.0814E+03	6.8499E+04

	ITERANT 1	REGION 1	REGION 2	REGION 3	REGION 4	REGION 5	REGION 6	REGION 7
GROUP 5...	3.9871E+07	5.8543E+03	0.	8.1361E+05	7.4130E+04	1.7872E+02	2.1450E+03	3.8418E+04
GROUP 6...	3.6454E+08	6.3833F+02	0.	7.4154E+05	7.8036E+04	1.3412E+02	3.7668E+02	1.3499E+04
GROUP 7...	3.9628E+07	3.7502E+00	0.	3.5543E+04	2.6047E+03	2.0646E+01	2.5147E+00	3.5218E+02
GROUP 8...	7.9277E+07	1.5312E-01	0.	9.7181E+03	1.6618E+03	4.1581E+01	1.8079E+01	7.6659E+01
GROUP 9...	8.3204E+07	4.5115E-04	0.	1.8214E+03	1.9553E+02	1.3592E-04	1.9996E-03	1.0399E+01
GROUP 10...	8.2904E+07	4.0476F-07	0.	4.7798E+01	9.3432E+00	8.2144E-02	1.8024E-05	8.0674E-01
GROUP 11...	2.8943E+07	5.2645E-10	0.	3.0266E+00	5.7751E-01	1.2243E-02	4.7245E-07	1.9624E-02
GROUP 12...	1.7252E+07	3.1628E-14	0.	7.4703E-03	3.3980E-03	1.5861E-05	9.1404E-10	8.8571E-06
GROUP 13...	8.0437E+07	1.9565E-27	0.	5.7399E-09	6.9630E-09	3.4901E-11	9.4635E-16	5.9495E-12
RAD(TIS)/HR	7.0624E+03	8.0679E+00	0.	6.1521E+01	7.0189E+00	4.5378E-02	1.8880E+00	2.3487E+01
RAD(ETH)/HR	9.8050E+03	1.1645E+01	0.	8.5980E+01	9.8944E+00	6.3380E-02	2.7294E+00	3.3706E+01
REM/HOUR	6.5349E+04	5.2564F+01	0.	4.4902E+02	5.0412E+01	3.0959E-01	1.2337E+01	1.5454E+02

************************NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS************************

	REGION 8	REGION 12	REGION 13	REGION 14	REGION 15	SCATTER 0	SCATTER 1	SCATTER 2
GROUP 1...	1.8383E+04	4.7857E+03	1.4849E+03	8.0035E+04	6.0340E+05	8.8750E+05	7.3522E+05	3.0527E+05
GROUP 2...	2.5377E+04	5.1326E+03	4.0191E+04	1.1077E+06	6.3956E+06	6.1623E+05	4.4950E+06	1.5856E+06
GROUP 3...	2.8657E+04	3.6268E+03	1.1052E+06	1.2189E+07	2.4489E+07	1.1771E+05	1.0964E+07	8.5107E+06
GROUP 4...	9.6821E+03	8.3598E+02	1.6573E+06	1.1542E+07	6.2324E+06	1.0200E+04	1.2889E+05	1.7964E+06
GROUP 5...	7.3100E+03	3.5484E+02	4.2268E+06	3.1102E+07	3.5980E+06	2.5005E+03	2.6451E+05	1.9476E+06
GROUP 6...	3.9409E+03	9.8509E+01	3.0643E+07	1.2204E+08	2.1102E+08	1.8641E+02	7.9907E+05	3.3742E+06
GROUP 7...	1.0734E+02	1.9387E+00	8.6751E+06	2.9411E+07	1.5037E+06	6.1613E-01	3.9155E+04	3.9441E+05
GROUP 8...	2.6538E+01	3.2700E-01	1.0141E+07	2.4345E+07	4.4781E+07	2.0950E-02	2.9579E+04	3.0962E+05
GROUP 9...	3.6849E+00	3.4883E-02	1.2925E+07	2.3291E+07	4.6986E+07	5.9251E-05	3.4275E+04	4.3087E+05
GROUP 10...	9.4981E-02	5.3207E-04	6.5236E+06	5.9885E+06	7.0392E+07	4.4951E-08	9.6463E+03	1.2820E+05
GROUP 11...	4.2501E-03	2.6677E-05	6.3433E+06	6.2469E+06	1.6352E+07	6.0304E-11	1.0569E+04	1.6457E+05
GROUP 12...	1.0756E-05	5.4496E-08	3.5215E+06	3.1613E+06	1.0569E+07	3.7495E-15	4.2042E+03	6.8539E+04
GROUP 13...	7.4304E-12	3.0346E-14	2.0265E+06	7.5835E+07	2.5753E+06	2.1852E-28	2.6555E+03	5.3467E+04
RAD(TIS)/HR	1.5001E+00	2.5096E-01	6.6248E+02	2.5970E+03	3.6704E+03	2.8716E+01	2.8214E+02	2.6128E+02
RAD(ETH)/HR	2.1448E+00	3.6285E-01	9.2217E+02	3.6095E+03	5.0844E+03	4.2290E+01	3.9418E+02	3.6337E+02
REM/HOUR	1.0110E+01	1.6440E+00	6.3853E+03	2.3693E+04	3.4355E+04	1.8468E+02	1.8960E+03	1.9094E+03

************************NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS************************

	SCATTER 3	SCATTER 4	SCATTER 5	SCATTER 6	SCATTER 7	SCATTER 8	ANGULAR 1	SPATIAL 1
GROUP 1...	1.4198E+05	8.1786E+04	2.1542E+03	2.5837E+02	1.9107E+01	6.9572E+00	2.1106E+06	2.1267E+06
GROUP 2...	1.5214E+06	1.0852E+06	3.4913E+04	2.8711E+04	3.8854E+03	2.1031E+03	8.9620E+06	9.6604E+06
GROUP 3...	9.8302E+06	3.7396E+06	1.7153E+06	3.2232E+06	4.6816E+05	1.6814E+06	3.7189E+07	4.3244E+07
GROUP 4...	6.4712E+06	1.9331E+06	2.4163E+06	5.3219E+06	1.4742E+06	5.0798E+04	1.7676E+07	2.3111E+07
GROUP 5...	1.7303E+07	2.2566E+06	8.1941E+05	1.5129E+07	8.6028E+05	2.8532E+05	3.3925E+07	4.6546E+07
GROUP 6...	5.4909E+07	6.5222E+06	2.0924E+06	8.3846E+07	5.4673E+06	9.7477E+05	3.3846E+08	4.1287E+08
GROUP 7...	9.6559E+06	1.3693E+06	4.4096E+05	2.5308E+07	1.7603E+06	1.9039E+05	3.4648E+07	4.6638E+07
GROUP 8...	6.3669E+06	9.8255E+05	2.9981E+05	2.3610E+07	2.1018E+06	2.5023E+05	7.4389E+07	8.9989E+07

```
SAMPLE PROBLEM 2, NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY   *****THE FASTER CODE******CASE    2
EMPTY PROPELLANT TANK, REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    8

GROUP    9... 6.1157E+06  1.1127E+06  5.2258E+05  2.1237E+07  2.7156E+06  3.9784E+05  7.8168E+07  9.4681E+07
GROUP   10... 1.7787E+06  3.1580E+05  1.3075E+05  6.4883E+06  7.5750E+05  1.1898E+05  8.0415E+07  9.1955E+07
GROUP   11... 2.3823E+06  4.0064E+05  1.3888E+05  4.1991E+06  1.1307E+06  5.4588E+05  2.7280E+07  3.2738E+07
GROUP   12... 7.0017E+05  9.5088E+05  1.1987E+05  1.4309E+06  2.9012E+05  5.9487E+04  1.6450E+07  1.9414E+07
GROUP   13... 5.8481E+05  5.7373E+07  7.9705E+06  5.6156E+06  5.3729E+06  2.0903E+05  6.9984E+07  9.9830E+07
 RAD(TIS)/HR 1.1822E+03  3.0003E+02  1.1302E+02  1.6404E+03  1.5000E+02  4.9203E+01  6.5119E+03  7.9730E+03
 RAD(ETH)/HR 1.6421E+03  4.2143E+02  1.5776E+02  2.2738E+03  2.0926E+02  6.7997E+01  9.0401E+03  1.1069E+04
    REM/HOUR 1.0392E+04  2.4640E+03  6.9632E+02  1.5578E+04  1.3835E+03  3.9158E+02  6.0355E+04  7.3915E+04
```



```
SAMPLE PROBLEM 2, NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY   *****THE FASTER CODE******CASE    2
EMPTY PROPELLANT TANK, REFERENCE ER-8236 (LOCKHEED STUDY FOR NASA-MSFC) *****T.M.JORDAN,WANL******PAGE    9

**********************BOUNDARY SEARCH PARAMETERS, (SURFACE,MOST PROBABLE NEXT REGION)**********************
REGION    1  (  -3,  2)  (   4,  4)  (  16,  3)
REGION    2  (  -2,  9)  (   3,  0)  (  16,  3)
REGION    3  (  -2,  9)  (   4,  4)  ( -16,  1)  (  17, 10)
REGION    4  (  -4,  1)  (   5,  5)  (  17, 10)
REGION    5  (  -5,  4)  (   6,  6)  (  17, 10)
REGION    6  (  -6,  5)  (   7,  7)  (  17, 10)
REGION    7  (  -7,  6)  (   8,  8)  (  17, 10)
REGION    8  (  -8,  7)  (   9, 11)  (  17, 10)
REGION    9  (  -1, 31)  (   2,  0)  (  20, 31)
REGION   10  ( -17,  7)  (  20, 31)  (  -2,  9)  (   9, 11)
REGION   11  (  -9, 10)  (  10, 12)  (  20, 31)
REGION   12  ( -10, 11)  (  11, 15)  ( -21,  0)
REGION   13  ( -10,  0)  (  18, 14)  (  21, 16)  ( -22, 18)
REGION   14  (  14,  0)  ( -18, 17)  (  19, 19)  ( -22,  0)  ( -10,  0)
REGION   15  ( -11, 12)  (  12, 16)  ( -21, 13)
REGION   16  ( -12, 15)  (  13, 17)  (  18, 14)  ( -21, 13)
REGION   17  ( -13, 16)  (  14, 20)  (  18, 14)
REGION   18  ( -10, 11)  (  19, 19)  (  22, 13)
REGION   19  ( -19,  0)  (  20, 31)  ( -10, 11)  (  14, 20)
REGION   20  ( -14,  0)  (  15, 31)  (  20, 31)
```

by section, and terminates at page 4 of case 1 (upper right hand corner of the printout). Every page after the first has the title cards printed at the top of the page.

Flux Output

The next page, (5), of the printout starts the output of the computed fluxes by detector and energy group. The various calculated flux components are separated by lines containing asterisks.

The first set of output, labeled: ** FLUXES FOR DETECTOR XXX AFTER XXX PACKETS **, is always printed first. (The term packet is just another way of describing an outer iteration; i.e., each energy dependent angular point source can be visualized as a packet of particles of different energies.) Each of the columns contains groupwise information as indicated on the left side of the printout. Note: there are less flux groups than were used in running the problem; i.e., 23 groups were collapsed to 11, as shown on Card 5-2. The first column is the coefficient of variation of the scalar flux, expressed as a fraction. The second column is the average energy in Mev. The third column is the number flux in particles/ cm^2 sec unless the input data were juggled; e.g., using unit areas and volumes for surface and volume detectors. The fourth column contains the energy flux in Mev/cm^2 sec. The fifth and sixth columns contain the group averaged differential number and energy fluxes, i.e., columns 3 and 4, respectively, divided by the group width. The last two columns are running summations of the fluxes and give the total flux from particles with energies greater than the lower boundary of the group.

The next set of output is labeled ** NUMBER FLUX RESPONSES FOR DETECTOR XXX AFTER XXX PACKETS **. Only one response function was input and its title (entered on Card 5-3) appears over the first column. The response by group is then printed. The last three lines in this column are the total response, the coefficient of variation as if the groupwise responses were obtained independently (labeled MIN ERROR), and the coefficient of variation as if the group wise responses were strictly dependent (labeled MAX ERROR).

This set of response function output would be omitted if no response functions were input. It would be repeated, until all response functions were output, for problems with more than 8 input responses.

The final set of output is labeled ** NUMBER FLUX MOMENTS FOR DETECTOR XXX AFTER XXX PACKETS **. Column 1 is always obtained in this set of output. It contains the average fluxes obtained between printouts. For this problem it is identical to the scalar flux. For problems with multiple printouts during the calculation, say every 100 outer iterations, this column would contain the average scalar flux as computed during the most recent 100 iterations. Subsequent columns contain other components of the total flux in the following order:

a) total flux by source, labeled SOURCE XXX (not used in this problem since only 1 source was present)

b) total scattered flux by scattering region, labeled REGION XXX (obtained in this problem for every non-void region)

c) total flux by order-of-scatter, labeled SCATTER XXX (obtained through the 7th scatter for this problem)

d) Legendre moments of the angular flux, labeled ANGULAR XXX (the current was obtained for this problem)

e) length-of-flight moments of the flux, labeled SPATIAL XXX (first moment obtained for this problem)

After printing the contribution by group, each column contains the total contribution to each of the response functions, if any.

The final page of printout for a problem (page 7 of this printout) is sometimes helpful in correcting geometric errors. It contains a listing, by region, of the bounding surfaces (with the sign of the ambiguity index affixed) and the region entered the last time a ray crossed these boundaries (most-probable-next-regions). Most-probable-next-region indices greater than the total number of regions (greater than 20 for this problem) indicate that there was no next region and should correspond to the outer boundaries of the problem. A zero indicates the boundary was never crossed.

The remainder of this printout, labeled CASE 2, is the neutron problem which was run as a change case immediately behind the first case. Most of the input changes involve the multigroup scattering cross sections. The flux printout (starting on page 6 of case 2) is interpreted in exactly the same manner as case 1.

The results contained in this printout are summarized in Table 3 along with appropriate results from Reference 20. Since the FASTER calculation was intended as a sample problem, comments on the relative accuracy and computer time requirements of the various calculations are not made.

TABLE 3

SUMMARY OF SAMPLE PROBLEM RESULTS

	FASTER Monte Carlo	18-0 * Monte Carlo	Point Kernel
Photons (rad/hr)	$3.4 \times 10^3 \pm 0.34 \times 10^3$	$\sim 6.7 \times 10^3 \pm 3.0 \times 10^3$	$\sim 3.0 \times 10^3$
Neutrons (rad/hr)	$7.1 \times 10^3 \pm 2.7 \times 10^3$	$\sim 2.4 \times 10^4 \pm 1.5 \times 10^4$	$\sim 2.0 \times 10^2$ (14-0, A.W.) $\sim 5.0 \times 10^2$ (QAD, H_2O) $\sim 6.6 \times 10^2$ (QAD, C)

*Reference 20, Figures 30 and 55

Astronuclear Laboratory

SECTION

11.0 REFERENCES

1. T. M. Jordan, "Reactor Shielding Calculations by the Monte Carlo Method," Douglas Paper No. 3271, May 1965, (Published in part in "Proceedings of the Conference on the Applications of Computing Methods to Reactor Problems," ANL 7050, August 1965)
2. T. M. Jordan, "Energy Independent Biasing Functions for Monte Carlo Transport Calculations," Douglas Paper No. 3636, November 1965, (Summarized in Transactions of the American Nuclear Society, November 1965)
3. T. M. Jordan, "SOBER, A Monte Carlo Program for Neutron and Gamma Ray Transport in Complex Geometries," DAC-60659, May 1967.
4. H. Goldstein, "Fundamental Aspects of Reactor Shielding," Addison Wesley Publishing Company, Inc., 1959.
5. Marginau and Murphy, "The Mathematics of Physics and Chemistry," D. Van Nostrand Company, Inc., 1956.
6. H. Goldstein and J. E. Wilkins, Jr., "Calculations of the Penetration of Gamma Rays," NYO 3075, June 30, 1954.
7. S. Wilks, "Mathematical Statistics," John Wiley and Sons, Inc., 1962.
8. M. H. Kalos, "On the Estimation of Flux at a Point by Monte Carlo," Nuclear Science and Engineering, May 1963.
9. J. J. Loechler, J. E. MacDonald, "Flexible Monte Carlo Programs FMC-N, FMC-G," APEX 706.
10. T. M. Jordan, "QUAD, A Computer Subroutine for Ray Tracing in Quadric Surface Geometries," Douglas Report SM-46333, December 1964.
11. Volume 6 of this report; WANL-PR(LL)010.
12. Volume 7 of this report; WANL-PR(LL)010.
13. Volume 8 of this report; WANL-PR(LL)010.

14. G. D. Joanou and J. S. Dudek, "GAM-1: A Consistent P_1 Multigroup Code for the Calculation of Fast Neutron Spectra and Multigroup Constants," GA 1850, June 1961.

15. E. T. Whittaker and G. N. Watson, "A Course of Modern Analysis," Cambridge Press, 1962.

16. "*IBM 7090/7094 Programming Systems Fortran IV Language*," C23-6274-1, May 1963.

17. "Synthesis of Calculational Methods for the Design and Analysis of Radiation Shields for Nuclear Rocket Systems--Midterm Presentation," WANL-TME-1597, March 7, 1967.

18. Volume 1 of this report; WANL-PR(LL)010.

19. Appendix, Volume 1 of this report; WANL-PR(LL)010. (C-RD)

20. "Evaluation of Methods for Computing Nuclear Rocket Radiation Fields," ER-8236, Lockheed-Georgia Company.

APPENDIX A

ALTERNATE MONTE CARLO PROCEDURES

Section 2.0 of this volume pertained to a development of the Monte Carlo method which utilized random sampling for all of the spatial integrations. This method is used in the FASTER program. It is not necessarily the most efficient procedure since there are many alternatives. An alternate technique for integrating the spatial dependence of the distributed fixed source is discussed below. The discussion is limited to 1) a technique which further reduces the integrations performed by random sampling and, 2) the consideration of a single point detector.

Alternate Uncollided Angular Flux Estimator

Equation for the uncollided scalar flux can be manipulated as

$$\phi_o(\vec{r}, E) = \iint_{4\pi} \left\{ \frac{\phi_o(\vec{r}, \vec{\Omega}, E)}{q^*(\vec{\Omega})} \right\} q^*(\vec{\Omega})\, d\Omega \tag{A.1}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \phi_o^*(\vec{r}, \vec{\Omega}_i, E) \tag{A.2}$$

where

$$\phi_o^*(\vec{r}, \vec{\Omega}_i, E) = \frac{1}{q^*(\vec{\Omega}_i)} \phi_o(\vec{r}, \vec{\Omega}_i, E) \tag{A.3}$$

$$\vec{\Omega}_i \text{ selected at random from } q^*(\vec{\Omega}) \tag{A.4}$$

$$\left.\begin{array}{l} q^*(\vec{\Omega}) \geq 0 \\ q^*(\vec{\Omega}) > 0 \quad \text{if} \quad \int_o^\infty \phi_o(\vec{r}, \vec{\Omega}, E)\, dE > 0 \\ \iint_{4\pi} q^*(\vec{\Omega})\, d\Omega = 1 \end{array}\right\} \tag{A.5}$$

Thus the angular flux is approximated for finite n by:

$$\phi_o^*(\vec{r},\vec{\Omega},E) = \frac{1}{n}\sum_{i=1}^{n}\phi_o^*(\vec{r},\vec{\Omega}_i,E)\,\delta(\vec{\Omega}-\vec{\Omega}_i) \qquad (A.6)$$

But $\phi_o^*(\vec{r},\vec{\Omega}_i,E)$ is given by a simple line integral, i.e., equation 2.2:

$$\phi_o^*(\vec{r},\vec{\Omega}_i,E) = \frac{1}{q^*(\vec{\Omega}_i)}\int_0^{\infty} S_o(\vec{r}-s\vec{\Omega}_i,\vec{\Omega}_i,E)\exp\left[-\int_0^s \Sigma^t(r-s'\vec{\Omega}_i,E)\,ds'\right]ds \qquad (A.7)$$

This integration can be performed numerically with extreme accuracy for non-void source regions with slowly varying source distributions by using the transformation:

$$du = \exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega}_i,\bar{E})\,ds'\right]ds \qquad (A.8)$$

i.e., the discrete values of s corresponding to discrete values of u will be exponentially distributed with approximately equal contributions to the integration. Note: the transformation is performed for some average energy $\bar{E}$ thus yielding the same "source" points for all energies. Solving A.8 for ds and substituting into A.7 gives the transformed equation.

$$\phi_o^*(\vec{r},\vec{\Omega}_i,E) = \frac{1}{q^*(\vec{\Omega}_i)}\int_u S_o(\vec{r}-s(u)\vec{\Omega}_i,\vec{\Omega}_i,E)\exp\left[+\int_0^{s(u)}\left\{\Sigma^t(\vec{r}-s'\vec{\Omega}_i,\bar{E})-\Sigma^t(\vec{r}-s'\vec{\Omega}_i,\bar{E})\right\}ds'\right]du \qquad (A.9)$$

where $s(u)$ is the solution for s of

$$u = \int_0^{s(u)}\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega}_i,\bar{E})\,ds'\right]ds \qquad (A.10)$$

This procedure can also be generalized to include the effects of the source distribution in the transformation. All of these considerations have been incorporated into the sampling function described in Section 8.3, which approximates equation 2.36. The only difference is: random

discrete values of s (or u) are obtained for performing the numerical integration.

Another procedure for computing the uncollided angular flux, is to perform all integrations numerically. When an angular integration is included in a rotated spherical coordinate system centered at the detector point, smaller computer times are expended than on most point kernel calculations. This is a result of the many "point sources" having the same "line-of-sight" to the detector; i.e., the geometric calculations are significantly reduced.

Alternate Single Scattered Flux Estimator

The use of the above technique does not yield a single point source which can be used to calculate the point representation of the scattered source at the first scattering point. However, the same technique can be applied at the first scattering point. The procedure, again, is to write the equation for the single scattered scalar flux. (This formalism isn't really necessary but it provides a convenient and consistent basis for the application of random sampling techniques. It is useful in inferring the form of optional sampling functions.)

$$\phi_1(\vec{r},E) = \iiint \left\{\frac{S_1(\vec{r};\vec{\Omega},E)\exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega},E)\,ds'\right]}{p_1^*(\vec{r}')\;s^2}\right\} p_1^*(\vec{r}')\,dV \qquad (A.11)$$

which yields a point, single-scattered source

$$W_1^s(\vec{\Omega},E) = \frac{1}{p_1^*(\vec{r}_1)}\,S_1(\vec{r}_1,\vec{\Omega},E) \qquad (A.12)$$

where $\vec{r}_1$ is selected at random from $p_1^*(\vec{r}')$ as shown in Section 2.4.

In Section 2.4 it was assumed that there was an energy dependent angular point source, $W_o(\vec{\Omega},E)$, at $\vec{r}_o$ which finally yielded $W_1^s(\vec{\Omega},E)$. The point source $W_o(\vec{\Omega},E)$ can be obtained at this point in the calculation although another technique is used for the uncollided flux. Nevertheless, other techniques can also be used. In particular, the technique discussed for the uncollided scalar flux can also be used for the single scattered source.

$$W_1^s(\vec{\Omega},E) = \frac{1}{p_1^*(\vec{r}_1)} \iint_{4\pi} \left\{ \frac{\int_0^\infty \phi_o(\vec{r}_1,\vec{\Omega}',E') \frac{d^2\Sigma^s}{d\Omega dE} (\vec{r}_1;\vec{\Omega}',E'\rightarrow\vec{\Omega},E)dE'}{q^*(\vec{\Omega}')} \right\} q^*(\vec{\Omega}')d\Omega' \qquad (A.13)$$

$$= \frac{1}{p_1^*(\vec{r}_1)} \frac{1}{q^*(\vec{\Omega}_1)} \int_0^\infty \phi_o(\vec{r}_1,\vec{\Omega}_1,E') \frac{d^2\Sigma^s}{d\Omega dE} (\vec{r}_1;\vec{\Omega}_1,E'\rightarrow\vec{\Omega},E)\, dE' \qquad (A.14)$$

where the equality holds in the sense of the expected value. Multiple scattered flux contributions can be obtained using the remainder of the techniques described in Section 2.4.

There is an implicit assumption in the concept of using a line integral along a fixed direction vector that the angular variations can be approximated with facility in $q^*(\vec{\Omega})$. Intuitively, this appears correct since the angular flux at a point detector usually varies slowly within the solid angle cone subtended by the source. Especially, fluxes at a point detector in the most intense portion of a source could be computed with accuracy using only a few discrete directions.

The amount of detail which can be built into sampling functions internal to a computer program is limited and is usually based on line-of-sight observations between source or scattering points and the detector point. For more difficult problems, where line-of-sight estimates are not strictly correct, i.e., where short circuiting around a shield actually yields the major component of the flux, the effects can be approximated by using several alternate definitions of $p_1^*(\vec{r}')$, where $p_1^*(\vec{r}')$ is used in randomly selecting the first-scatter points. This same capability is not usually warranted for multiple scattering events since the effect is harder to predict and control.

APPENDIX B

ANGULAR AND SPATIAL FLUX MOMENTS

Legendre moments of the angular flux are obtained by an integration of the individual flux contributions over solid angle. Using the notation of Section 7.0, the angular dependence of the flux contributions is denoted by:

$$\Delta\phi^*_{i,j,k}(\vec{\Omega}) = \Delta\phi_{i,j,k}\, \delta(\vec{\Omega} - \vec{\Omega}_o)$$

where $\Delta\phi_{i,j,k}$ is the flux contribution to the jth energy group from the kth inner iteration of the ith outer iteration and is obtained for the fixed direction $\vec{\Omega}_o$.

The angular moments are averaged over the azimuthal angle and obtained with respect to a preferred direction $\vec{\Omega}_p$:

$$\phi_j^l = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0} \iint_{4\pi} \Delta\phi_{i,j,k}\, \delta(\vec{\Omega} - \vec{\Omega}_o)\, P_l(\vec{\Omega}\cdot\vec{\Omega}_p)\, d\Omega \qquad (B.1)$$

$$= \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0} \Delta\phi_{i,j,k}\, P_l(\vec{\Omega}_o\cdot\vec{\Omega}_p) \qquad (B.2)$$

where

$$\left.\begin{aligned} P_o(\mu) &= 1 \\ P_1(\mu) &= \\ P_l(\mu) &= \frac{1}{l}\left[(2l-1)\,\mu P_{l-1}(\mu) - (l-1)\,P_{l-2}(\mu)\right] \end{aligned}\right\} \qquad (B.3)$$

The preferred direction $\vec{\Omega}_p$ is fixed by input for point and volume detectors. For surface detectors, it is the unit normal to the surface, $\vec{n}$, as defined by equation 4.11. The zeroth moment, $l = 0$, is the scalar flux as given by equation 7.6 and the first moment, $l = 1$, is the particle current with respect to $\vec{\Omega}_p$.

The azimuthally averaged Legendre moments can be used to define azimuthally averaged differential angular fluxes and/or interval integrated values. This is not done by the FASTER program. However, the equations required for external manipulation of these moments are summarized below:

a) differential angular flux

$$\phi_j(\mu) = \sum_{l=0} \frac{2l+1}{4\pi} \phi_j^l \, P_l(\mu) \quad (B.4)$$

b) interval integrated flux, $\mu_m \leq \mu \leq \mu_{m+1}$

$$\bar{\phi}_{j,m} = \sum_{l=0} \frac{2l+1}{2} \phi_j^l \int_{\mu_m}^{\mu_{m+1}} P_l(\mu)\, d\mu \quad (B.5)$$

Using the relationship (Reference 15, page 309):

$$P_l(\mu) = \frac{1}{2l+1} \left\{ \frac{dP_{l+1}(\mu)}{d\mu} - \frac{dP_{l-1}(\mu)}{d\mu} \right\} \quad (B.6)$$

Then

$$\bar{\phi}_{j,m} = \frac{1}{2} \sum_{l=0} \phi_j^l \left\{ P_{l+1}(\mu_{m+1}) - P_{l+1}(\mu_m) - P_{l-1}(\mu_{m+1}) + P_{l-1}(\mu_m) \right\} \quad (B.7)$$

Spatial Moments

Length-of-flight moments of the flux can also be obtained from FASTER. These are normalized to a reference distance t_o. Thus

$$\bar{\phi}_j^m = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0} \Delta\phi_{i,j,k} \left(\frac{t_{i,k}}{t_o} \right)^m \quad (B.8)$$

$$t_o = \max\left(|\vec{r}_p - \vec{r}_c|, 100 \right), \; \vec{r}_p, \vec{r}_c \text{ are defined in Section 8.3} \quad (B.9)$$

$$t_{i,k} = \sum_{k'=0}^{k} |\vec{r}_{i,k'} - \vec{r}_{i,k'-1}| + \Delta t_{i,k} \quad (B.10)$$

where

$$\Delta t_{i,k} = |\vec{r} - \vec{r}_{i,k}| \text{ for a point detector}$$
$$= s, \text{ the distance to a surface detector} \quad (B.11)$$

Moments of the average length-of-flight are obtained for volume detectors.

$$\Delta t_{i,k} = s + \Delta s/2 \quad \text{if void}$$
$$= s + 1/\Sigma_j^v - \Delta s / \left[\exp(\Sigma_j^v \Delta s) - 1 \right] \quad \text{if not void} \quad (B.12)$$

For photons and non-volume detectors, the length-of-flight moments can be related to the time of arrival of the flux

$$t_{i,k} = c\,\tau_{i,k}, \quad \tau_{i,k} = t_{i,k}/c \quad (B.13)$$

where c is the velocity of light and τ is the time. If the source is assumed to have a time dependence $\delta(\tau)$, then, the temporal moments are given by

$$\left.\begin{aligned} \hat{\phi}_j^m &= \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0} \Delta\phi_{i,j,k} \, \tau_{i,k}^m \\ &= \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0} \Delta\phi_{i,j,k} \left(\frac{t_{i,k}}{c} \right)^m \\ &= \left(\frac{t_o}{c} \right)^m \bar{\phi}_j^m \end{aligned}\right\} \quad (B.14)$$

These moments are not processed by the FASTER program but the obvious procedure is to use techniques of the type used for the spatial dependence in the moments[6]; e.g., assume a functional form and match moments.

$$\left.\begin{aligned} \phi_j(\tau) &= \exp\left[-a(\tau - \tau_a) \right] \sum_{m'=0} b_{m'} (\tau - \tau_a)^{m'}, \; \tau \geq \tau_a \\ &= 0, \; \tau < \tau_a \end{aligned}\right\} \quad (B.15)$$

where τ_a is the earliest arrival time, α is an exponential decay constant, and the b_m's are constants in a polynomial representation of the departure from a true exponential decay.

Then

$$\hat{\phi}_i^m = \int_{\tau_a}^{\infty} \phi_i(\tau) \, \tau^m \, d\tau$$

$$= \int_{\tau_a}^{\infty} \exp\left[- \alpha(\tau - \tau_a)\right] \sum_{m'=0} b_{m'} (\tau - \tau_a)^{m'} \tau^m d\tau$$

$$= \sum_{m'=0} b_{m'} \, c_{m,m'} \, , \; m = 0, 1, \ldots \tag{B.16}$$

where

$$c_{m,m'} = \int_{\tau_a}^{\infty} \exp\left[-\alpha(\tau - \tau_a)\right] \tau^m (\tau - \tau_a)^{m'} d\tau \tag{B.17}$$

Equation B.16 is solved for the constants $b_{m'}$, $m' = 0, 1, \ldots$ and equation B.15 is used to compute the temporal dependence.

APPENDIX C

PROGRAM LISTING

The FORTRAN IV listing of the FASTER program is given in this appendix. Compatibility with other computer facilities can be obtained by the following changes in the control program on the first page of the listing:

a) Change of input tape logical designation from 5 to i
   replace M1 = 5 by M1 = i
b) Change of output tape logical designation from 6 to j
   replace M2 = 6 by M2 = j
c) Change of maximum number of lines per printout page from 43 to k
   replace LINEX = 43 by LINEX = k
d) Change of maximum number of locations for dimensioned arrays from 12000 to l
   replace COMMON H(12000) by COMMON H(l)
   and replace NSTORE = 12001 by NSTORE = l + 1

The listing corresponds to an operational program for the IBM 7094 computer which uses a MAP random number generator as shown on the last page of the listing. For conversion to the CDC 6600 computer, the MAP routine is removed and a card in the control program *(first page of listing)* is changed:

replace IBMCDC = 0 by IBMCDC = 1

It is assumed that the random number generator, RANF, distributed by the Control Data Corporation is on the library tape and that the calling sequence is:

R = RANF (i), where $i > 0$, stores i as the generator
R = RANF (0) yields a random number R on (0, 1)

All calls to the random number generator are relayed through the function subprogram RANNO (n) where n is a do-nothing argument--n is not used in the calling subprograms and can be defined arbitrarily in RANNO. Therefore, any other random number generator can be used by FASTER with appropriate changes in RANNO.

FORTRAN IV LISTING IS PRESENTED ON PAGES 184 THROUGH 259.

```
$IBFTC FAST     M94/2,XR7
CFASTER*FORTRAN-ANALYTIC SOLUTION-TRANSPORT EQUATION-RANDOM SAMPLING*TMJFAST0001
      COMMON           H(12009)                                          FAST0002
      COMMON/CDCIBM/IBMCDC                                               FAST0003
      COMMON/TAPEID/M1      ,M2                                          FAST0004
      COMMON/CASEID/KASE   ,NPAGE ,LINES ,LINEX ,TITLEA(18)    ,TITLEB(18)FAST0005
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NMMAX ,NSTMAX,     FAST0006
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,     FAST0007
     2              NUNITX,NIMAX ,NMPAX ,NORDER,NDOWN ,INELAS,NTRANS,     FAST0008
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,     FAST0009
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPDMAX,NPDMOD,NSGMAX,     FAST0010
     5              NVOMAX,NVOMCD,NPOINT,MODELP,MODELQ,MODELU,MODELV,     FAST0011
     6              NPRINT,NUNITS,NUMCER,KALIDE                           FAST0012
      COMMON/INDEXS/INTP   ,IAZ    ,IISV   ,IMTL   ,IRHO   ,IX9    ,IELL   , FAST0013
     1              IFLW   ,IAE    ,IBE    ,INSG   ,INPC   ,IJSN   ,IJSX   , FAST0014
     2              IXTR   ,ISUV   ,IATN   ,IATW   ,IES3   ,ISSH   ,IDEN   , FAST0015
     3              INTG   ,IFLF   ,IFGW   ,IIDS   ,IIDV   ,IIDR   ,IIDS   , FAST0016
     4              IVOL   ,ICDF   ,IXOT   ,IRSI   ,IALP   ,IVMD   ,IGIM   , FAST0017
     5              IAIM   ,IALM   ,IALH   ,IA     ,INS    ,IVEE   ,IVAL   , FAST0018
     6              ISPK   ,ISPE   ,IATD   ,IRSP   ,ISGT   ,IIDE   ,IIDI   , FAST0019
     7              ISGS   ,NEXT   ,IXMP   ,ISGR   ,IWS    ,IFS    ,IWC    , FAST0020
     8              ICC    ,IND    ,IU     ,IV     ,IW     ,ISI    ,IST    , FAST0021
     9              INRG   ,INSC   ,ISTP   ,INRP   ,INCP   ,IFXP   ,IFXS   , FAST0022
     1              IFXT   ,IFXE   ,IFXA                                     FAST0023
      COMMON/OTHERS/RADIUS,XCT(3),DELTA  ,BOC(3),ATA    ,ATB    ,ATC    ,  FAST0024
     1              ATD    ,AT     ,BT     ,AS     ,BS     ,JMIN   ,JMAX   ,  FAST0025
     2              KMIN   ,KMAX   ,JBAR   ,NZERO                            FAST0026
      COMMON/REGSCA/NSRMAX,INSR                                          FAST0027
      IBMCDC = 0                                                         CORRECT
      NSTORE = 12001                                                     FAST0028
      M1 = 5                                                             FAST0029
      M2 = 6                                                             FAST0030
      LINEX = 43                                                         FAST0031
      KASE = 0                                                           FAST0032
   10 CALL DEFINE                                                        FAST0033
      CALL SOLVIT                                                        FAST0034
      GO TO 10                                                           FAST0035
      END                                                                FAST0036
$IBFTC LABLE    M94/2,XR7                                                FAST0037
CLABEL *OUTPUT PAGE HEADINGS FOR PROGRAM FASTER*T.M.JORDAN*WANL*OCT.1966FAST0038
      FUNCTION LABEL(LINE)                                               FAST0039
      COMMON/TAPEID/M1      ,M2                                          FAST0040
      COMMON/CASEID/KASE   ,NPAGE ,LINES ,LINEX ,TITLEA(18)    ,TITLEB(18)FAST0041
      L = IABS(LINE)                                                     FAST0042
      IF(LINE.LE.0)GO TO 10                                              FAST0043
      LINES = LINES + L                                                  FAST0044
      LABEL = 0                                                          FAST0045
                                                                         FAST0046
```



```
      IF(LINES.LE.LINEX)GO TO 20                                         FAST0047
   10 LINES = L + 3                                                      FAST0048
      NPAGE = NPAGE + 1                                                  FAST0049
      LABEL = 1                                                          FAST0050
      WRITE(M2,2000)TITLEA,KASE,TITLEB,NPAGE                             FAST0051
 2000 FORMAT(1H1,18A4,30H*****THE FASTER CODE******CASE,I5/1X,18A4,30H**FAST0052
     1***T.M.JORDAN,WANL******PAGE,I5/1H )                               FAST0053
   20 RETURN                                                             FAST0054
      END                                                                FAST0055
$IBFTC VECTR    M94/2,XR7                                                FAST0056
CVECTOR *UNIT VECTOR                                                     FAST0057
      FUNCTION VECTOR(X,Y,C)                                             FAST0058
      DIMENSION X(3),Y(3),C(3)                                           FAST0059
      FST = 0.0                                                          FAST0060
      DO 10 I=1,3                                                        FAST0061
      C(I) = Y(I) - X(I)                                                 FAST0062
   10 FST = FST + C(I)**2                                                FAST0063
      FST = SQRT(FST)                                                    FAST0064
      IF(FST.GT.0.0) GO TO 15                                            FAST0065
      C(1) = 0.0                                                         FAST0066
      C(2) = 0.0                                                         FAST0067
      C(3) = 1.0                                                         FAST0068
      GO TO 25                                                           FAST0069
   15 DO 20 I=1,3                                                        FAST0070
   20 C(I) = C(I)/FST                                                    FAST0071
   25 VECTOR = FST                                                       FAST0072
      RETURN                                                             FAST0073
      END                                                                FAST0074
$IBFTC BLANK    M94/2,XR7                                                FAST0075
CZEROUT*ZERO OUT ARRAY                                                   FAST0076
      SUBROUTINE ZEROUT(MAX,N)                                           FAST0077
      DIMENSION N(1)                                                     FAST0078
      IF(MAX.LE.0)GO TO 20                                               FAST0079
      DO 10 I=1,MAX                                                      FAST0080
   10 N(I) = 0                                                           FAST0081
   20 RETURN                                                             FAST0082
      END                                                                FAST0083
$IBFTC LOKATE  M94/2,XR7                                                 FAST0084
CLOCATE*REGION INDEX CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966FAST0085
      FUNCTION LOCATE(NNN,XXX)                                           FAST0086
      DIMENSION XXX(1)                                                   FAST0087
      COMMON           H(1)                                              FAST0088
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NMMAX ,NSTMAX,     FAST0089
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,     FAST0090
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,     FAST0091
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,     FAST0092
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPDMAX,NPCMOD,NSGMAX,     FAST0093
```

```
5                NVDMAX,NVOMCD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST0094
6                NPRINT,NUNITS,NUMBER,KALIDE                         FAST0095
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,INTL  ,IRHO  ,IXR   ,IELL  ,  FAST0096
1                IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST0097
2                IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST0098
3                INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST0099
4                IVOL  ,ICDT  ,IXCT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST0100
5                IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST0101
6                ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST0102
7                ISGS  ,NEXT  ,IXHP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST0103
8                IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST0104
9                INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST0105
1                IFXT  ,IFXE  ,IFXA                                  FAST0106
      COMMON/DUMLCC/NN    ,MM    ,X(9)                               FAST0107
      DO 10 I=1,3                                                    FAST0108
   10 X(I) = XXX(I)                                                  FAST0109
      NN = NNN                                                       FAST0110
      CALL LOCDUM(NAMAX,NBMAX,1,H(INTP),H(IAZ),H(IA),H(INS),H(IND),  FAST0111
   1            H(IU))                                               FAST0112
      LOCATE = MM                                                    FAST0113
      RETURN                                                         FAST0114
      END                                                            FAST0115
$IBFTC LOCOML   M94/2,XR7                                            FAST0116
CLOCDUM*REGION INDEX CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966FAST0117
      SUBROUTINE LOCDUM(L1,L2,L3,NTP,AZ,A,NS,ND,U)                   FAST0118
      DIMENSION NTP(1),AZ(1),A(L1,L3),NS(L2,L3),ND(1),U(1)           FAST0119
      COMMON/DUMLCC/NNN   ,NN    ,X(9)                               FAST0120
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX, FAST0121
1                NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST0122
2                NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST0123
3                NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST0124
4                NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST0125
5                NVDMAX,NVDMCD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST0126
6                NPRINT,NUNITS,NUMBER,KALIDE                         FAST0127
      DO 100 I=1,3                                                   FAST0128
      X(I+3) = X(I)*X(I)                                             FAST0129
      J = I + 1 - 3*(I/3)                                            FAST0130
  100 X(I+6) = X(I)*X(J)                                             FAST0131
      DO 110 I=1,NSMAX                                               FAST0132
  110 ND(I) = 0                                                      FAST0133
      DO 200 N=1,2                                                   FAST0134
      IF(N.GT.1)GO TO 130                                            FAST0135
      IMIN = NNN                                                     FAST0136
      IF((IMIN-1)*(NRMAX-IMIN+1).GT.0)GO TO 120                      FAST0137
      IMIN = NRMAX + 1                                               FAST0138
      GO TO 200                                                      FAST0139
  120 IMAX = NRMAX                                                   FAST0140
```



```
      GO TO 140                                                      FAST0141
  130 IMAX = IMIN - 1                                                FAST0142
      IMIN = 1                                                       FAST0143
  140 DO 190 I=IMIN,IMAX                                             FAST0144
      DO 170 J=1,NBMAX                                               FAST0145
      IF(NS(J,I).EQ.0)GO TO 180                                      FAST0146
      KP = NS(J,I)/1000000                                           FAST0147
      K = NS(J,I)/1000 - 1000*KP                                     FAST0148
      IF(ND(K).GT.0)GO TO 160                                        FAST0149
      ND(K) = 1                                                      FAST0150
      MAX = NTP(K)                                                   FAST0151
      U(K) = AZ(K)                                                   FAST0152
      DO 150 L=1,MAX                                                 FAST0153
  150 U(K) = U(K) + X(L)*A(L,K)                                      FAST0154
  160 IF(FLOAT(2*KP - 1)*U(K).GT.0.0)GO TO 190                       FAST0155
  170 CONTINUE                                                       FAST0156
  180 NN = I                                                         FAST0157
      GO TO 210                                                      FAST0158
  190 CONTINUE                                                       FAST0159
  200 CONTINUE                                                       FAST0160
      NN = 0                                                         FAST0161
  210 RETURN                                                         FAST0162
      END                                                            FAST0163
$ORIGIN        ALPHA                                                 FAST0164
$IBFTC DEFEIN  M94/2,XR7                                             FAST0165
CDEFINE*DATA INPUT AND PREPARATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*66FAST0166
      SUBROUTINE DEFINE                                              FAST0167
      COMMON       V(1)                                              FAST0168
      COMMON/TAPEID/M1    ,M2                                        FAST0169
      COMMON/CASEID/KASE  ,NPAGE ,LINES ,LINEX ,TITLEA(18)   ,TITLEB(18)FAST0170
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX, FAST0171
1                NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST0172
2                NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST0173
3                NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST0174
4                NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST0175
5                NVDMAX,NVDMCD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST0176
6                NPRINT,NUNITS,NUMBER,KALIDE                         FAST0177
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,INTL  ,IRHO  ,IXR   ,IELL  ,  FAST0178
1                IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST0179
2                IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST0180
3                INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST0181
4                IVOL  ,ICDT  ,IXCT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST0182
5                IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST0183
6                ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST0184
7                ISGS  ,NEXT  ,IXHP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST0185
8                IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST0186
9                INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST0187
```

```
1                IFXT  ,IFXE  ,IFXA                                      FAST0188
 COMMON/INPUTS/IN1    ,INN(23)                                           FAST0189
 DIMENSION TDUM(36),TYPE(5,6)          ,ADUM(18)                         FAST0190
 DATA TDUM/                                                              FAST0191
1   4H(F)O,4HRTRA,4HN,(A,4H)NAL,4HYTIC,4H (S),4HOLUT,4HION,,4H(T)R,FAST0192
2   4HANSP,4HORT ,4H(E)O,4HUATI,4HON,(,4HR)AN,4HDOM ,4HSAMP,4HLING,FAST0193
3   4HMATH,4HEMAT,4HICAL,4H AND,4H NUM,4HERIC,4HAL A,4HNALY,4HSIS,,FAST0194
4   4HCOMP,4HUTER,4H PRG,4HGRAM,4HMING,4H AND,4H CHE,4HCKOU,4HT BY/FAST0195
 DATA TYPE/                                                              FAST0196
1   4H VAR,4HIABL,4HE DI,4HMENS,4HIONS,                                  FAST0197
2   4HSURF,4HACES,4H AND,4H REG,4HIONS,                                  FAST0198
3   4H IND,4HEPEN,4HDENT,4H SOU,4HRCES,                                  FAST0199
4   4HBASI,4HC CR,4HOSS ,4HSECT,4HIONS,                                  FAST0200
5   4HDETE,4HCTOR,4HS + ,4HRESP,4HONSE,                                  FAST0201
6   4H SAM,4HPLIN,4HG PA,4HRAME,4HTERS/                                  FAST0202
10 KASE = KASE + 1                                                       FAST0203
   NPAGE = 0                                                             FAST0204
   LINES = LINEX                                                         FAST0205
   IF(KASE.GT.1) GO TO 30                                                FAST0206
   DO 20 J=1,36                                                          FAST0207
20 TITLEA(J) = TDUM(J)                                                   FAST0208
30 CONTINUE                                                              FAST0209
   DO 700 NNN=1,6                                                        FAST0210
   IF(LABEL(1).GE.0) WRITE(M2,2000) NNN,(TYPE(I,NNN),I=1,5)              FAST0211
2000 FORMAT(1X,32(1H*),18HDATA INPUT SECTION,I2,2H, ,5A4,33(1H*))        FAST0212
   CALL READI(24,IN1)                                                    FAST0213
   IF(IN1.LE.0) GO TO 60                                                 FAST0214
   DO 50 I=1,IN1                                                         FAST0215
50 CALL READA(18,ADUM)                                                   FAST0216
60 GO TO (100,200,300,400,500,600), NNN                                  FAST0217
100 CALL STORER                                                          FAST0218
   GO TO 700                                                             FAST0219
200 CONTINUE                                                             FAST0220
   CALL GEOMIN(NAMAX,NBMAX,1,V(INTP),V(IAZ),V(IA),V(ISV),V(IMTL),        FAST0221
  1             V(INS),V(IRHO),V(IXR))                                   FAST0222
   GO TO 700                                                             FAST0223
300 IE   = NSTORE - NXEMAX                                               FAST0224
   IEN  = IE   - NXEMAX                                                  FAST0225
   IENG = IEN  - NXEMAX                                                  FAST0226
   IEBG = IENG - NXEMAX                                                  FAST0227
   IF(IEBG.LT.NEXT) NERROR = NERROR + 1                                  FAST0228
   CALL SOURCE(NEMAX,NXMAX,5,1,V(IELL),V(IELW),V(IAE),V(IBF),V(INSG),FAST0229
  1             V(INPC),V(IJSN),V(IJSX),V(ISUV),V(IXTR),V(IVEF),         FAST0230
  2             V(IVAL),V(ISPW),V(ISPE),V(ISV),V(IE  ),V(IEN),           FAST0231
  3             V(IEBG),V(IENG))                                         FAST0232
   GO TO 700                                                             FAST0233
400 IEAC = NSTORE - NEMOD*(NIMAX + NMMAX)                                FAST0234
```



```
   IJMX = IEAC - NXSECT*(INELAS + 3)                                     FAST0235
   IXSI = IJMX - NXSECT*INELAS*NTRANS                                    FAST0236
   IXSE = IXSI - NXSECT* NDOWN*NEMAX*MAX0(2,NORDER)                      FAST0237
   IXST = IXSE - NEMOD                                                   FAST0239
   IN1  = IXST - ISGS                                                    FAST0240
   CALL INSECT(NEMOD,NDOWN,NORDER,INELAS,NEMAX,1,V(IESB),V(IDEN),        FAST0241
  1             V(ISSH),V(ISGT),V(IDE),V(IDI)          ,V(ISGS),         FAST0242
  2             V(IMTL),V(IRHO),V(IXST),V(IATO),V(IJMX),V(IXSE),         FAST0243
  3V(IXSI),V(IEAC),V(IELL),NTRANS)                                       FAST0244
   NEXT = ISGS + NNMAX                                                   FAST0245
   IF(NEXT.GT.IXMP) NERROR = NERROR + 1                                  FAST0246
   GO TO 700                                                             FAST0247
500 CALL RESULT(NGMOD,1,V(INTG),V(IELF),V(IFGW),V(ITDS),V(IRSP),         FAST0248
  1             V(IDR),V(IDS),V(IVOL),V(ICDT),V(IXDT),V(IELL),           FAST0249
  2V(IMTL),V(IRHO),V(IEAC),NEMOD)                                        FAST0250
   GO TO 700                                                             FAST0251
600 CALL RANDOM(V(IRSI),V(IALP),V(IVMD),V(IGIM),V(IAIM),V(IALM),         FAST0252
  1V(IALH),V(INPC),V(IVEE),NXMAX,5)                                      FAST0253
700 CONTINUE                                                             FAST0254
   IF(LABEL(1).GE.0)WRITE(M2,2010)                                       FAST0255
2010 FORMAT(1X,35(1H*),36HDATA INPUT AND PREPARATION COMPLETED,36(1H*))FAST0256
   IF(NERROR.EQ.0) GO TO 900                                             FAST0257
   IF(LABEL(1).GE.0)WRITE(M2,2020)                                       FAST0258
2020 FORMAT(1X,42(1H*),23HPROCEEDING TO NEXT CASE,42(1H*))               FAST0259
   GO TO 10                                                              FAST0260
900 MAX = NSTORE - 1                                                     FAST0261
   DO 910 I=NEXT,MAX                                                     FAST0262
910 V(I) = 0.0                                                           FAST0263
   IF(LABEL(1).GE.0)WRITE(M2,2040)                                       FAST0264
2040 FORMAT(1X,1H*,35(3HGO*))                                            FAST0265
   LINES = LINEX + 1                                                     FAST0266
   RETURN                                                                FAST0267
   END                                                                   FAST0268
$IBFTC STORE    M94/2,XR7                                                FAST0269
CSTORER*LIMIT INPUT AND VARIABLE DIMENSIONING FOR PROGRAM FASTER*THJ*66*FAST0270
   SUBROUTINE STORER                                                     FAST0271
   COMMON        N(1)                                                    FAST0272
   COMMON/REGSCA/NSRMAX,INSR                                             FAST0273
   COMMON/TAPEID/M1     ,M2                                              FAST0274
   COMMON/CASEID/KASE  ,NPAGE ,LINES ,LINEX ,TITLEA(18)   ,TITLEB(18)FAST0275
   COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,       FAST0276
  1             NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,        FAST0277
  2             NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,        FAST0278
  3             NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,        FAST0279
  4             NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,        FAST0280
  5             NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,        FAST0281
  6             NPRINT,NUNITS,NUMBER,KALIDE                              FAST0282
```

190

```
      COMMON/INDEXS/INTP   ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST0283
     1               IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST0284
     2               IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST0285
     3               INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST0286
     4               IVOL  ,ICDT  ,IXDT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST0287
     5               IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST0238
     6               ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST0288
     7               ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST0289
     8               IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST0290
     9               INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST0291
     1               IFXT  ,IFXE  ,IFXA                                  FAST0292
      COMMON/INPUTS/IN1    ,IN2   ,IN3   ,IN4   ,IN5   ,IN6   ,IN7   ,   FAS10293
     1               IN8   ,IN9   ,IN10  ,IN11  ,INN(13)                 FAST0294
      DIMENSION LCP(51),ARRAY(73),LOC(1)                                 FAST0295
      EQUIVALENCE (LOC(1),INTP)                                          FAST0296
      EQUIVALENCE(LCP(39),L1),(LCP(40),L2),(LCP(41),L3),(LCP(42),L4),   FAST0297
     1(LCP(43),L5),(LCP(44),L6),(LCP(45),L7),(LCP(46),L8),(LCP(47),L9), FAST0298
     2(LCP(48),L10),(LCP(49),L11),(LCP(50),L12)                          FAST0299
      DATA  ARRAY          / 3HNTP,3HAZ ,3HISV,3HMTL,3HRHO,3HXR ,3HELL,FAST0300
     13HELW,3HAE ,3HBE ,3HNSG,3HNPC,3HJSN,3HJSX,3HXTR,3HSUV,3HATN,3HATW,FAST0301
     23HESB,3HSSH,3HDEN,3HNTG,3HELF,3HFGW,3HTDS,3HIDV,3HIDR,3HIDS,3HVOL,FAST0302
     33HCDT,3HXDT,3HRSI,3HALP,3HVMD,3HGIM,3HAIM,3HALM,3HALH,3HA  ,3HNS ,FAST0303
     43HVEE,3HVAL,3HSPW,3HSPE,3HATD,3HRSP,3HSGT,3HIDE,3HIDI,3HSGS,3HEND,FAST0304
     53HXMP,3HSGR,3HWS ,3HES ,3HWC ,3HEC ,3HND ,3HU  ,3HV  ,3HW  ,3HSI ,FAST0305
     63HST ,3HNRG,3HNSC,3HSTP,3HNRP,3HNCP,3HFXP,3HFXS,3HFXT,3HFXE,3HFXA/FAST0306
      IF(KASE.GT.1) GO TO 20                                             FAST0307
      NERROR = 0                                                         FAST0308
      NNMAX = 0                                                          FAST0309
      MAX = NSTORE - 1                                                   FAST0310
      DO 10 I=1,MAX                                                      FAST0311
   10 N(I) = 0                                                           FAST0312
      GO TO 30                                                           FAST0313
   20 MAXS = NSMAX                                                       FAST0314
      MAXA = NAMAX                                                       FAST0315
      MAXR = NRMAX                                                       FAST0316
      MAXB = NBMAX                                                       FAST0317
      MAXV = NVMAX                                                       FAST0318
      MAXX = NXMAX                                                       FAST0319
      MAXE = NEMAX                                                       FAST0320
      MAXM = NMMAX                                                       FAST0321
      MAXI = NIMAX                                                       FAST0322
      MAXF = NFMAX                                                       FAST0323
      MODG = NGMOD                                                       FAST0324
      MODE = NEMOD                                                       FAST0325
      MODX = NXSECT                                                      FAST0326
      MODO = NORDER                                                      FAST0327
      MODD = NDOWN                                                       FAST0328
```

191

```
      MODI = INELAS                                                      FAST0329
      MAXN = NNMAX                                                       FAST0330
   30 IF(IN2 .GT.0) CALL READA(18,TITLEA)                                FAST0331
      IF(IN3 .GT.0) CALL READA(18,TITLEB)                                FAST0332
      IF(IN4 .GT.0) CALL READI( 5,NSMAX )                                FAST0333
      IF(IN5 .GT.0) CALL READI( 4,NEMAX )                                FAST0334
      IF(IN6 .GT.0) CALL READI( 5,NXSECT)                                FAST0335
      IF(IN7 .GT.0) CALL READI( 4,NORDER)                                FAST0336
      IF(IN8 .GT.0) CALL READI( 8,NGMAX )                                FAST0337
      IF(IN8.GT.0) NSRMAX = NGMOD                                        FAST0338
      IF(IN9 .GT.0) CALL READI( 5,NPCINT)                                FAST0339
      IF(IN10.GT.0) CALL READI( 4,NPRINT)                                FAST0340
      IF(KASE.EQ.1) GO TO 70                                             FAST0341
      NARRAY = 51                                                        FAST0342
      IF(NXSECT.EQ.MODX) GO TO 40                                        FAST0343
      NARRAY = 47                                                        FAST0344
      NNMAX = 0                                                          FAST0345
   40 MOVE = LOC(NARRAY)                                                 FAST0346
      NMOVE = NSTORE - MOVE                                              FAST0347
      DO 50 I=1,NARRAY                                                   FAST0348
   50 LCP(I) = LOC(I) + NMOVE                                            FAST0349
      I = NSTORE                                                         FAST0350
      J = MOVE                                                           FAST0351
      NMOVE = MOVE - 1                                                   FAST0352
      DO 60 K=1,NMOVE                                                    FAST0353
      I = I - 1                                                          FAST0354
      J = J - 1                                                          FAST0355
   60 N(I) = N(J)                                                        FAST0356
   70 NEMOD = NEMAX + 1                                                  FAST0357
      NGMOD = NGMAX + 1                                                  FAST0358
      MOMENT = NVMOD + NCMAX + NLMAX + NTMAX + 1                         FAST0359
      MOMENT = MOMENT + NSRMAX                                           FAST0360
      NOMOD = 1                                                          FAST0361
      IF(NPOINT.EQ.0) NOMOD = NOMAX                                      FAST0362
      INTP = NSRMAX + 1                                                  FAST0363
      IAZ  = INTP + NSMAX                                                FAST0364
      IISV = IAZ  + NSMAX                                                FAST0365
      IMTL = IISV + NRMAX                                                FAST0366
      IRHO = IMTL + NRMAX                                                FAST0367
      IXR  = IRHO + NRMAX                                                FAST0369
      IELL = IXR  + NRMAX*3                                              FAST0370
      IELW = IELL + NEMOD                                                FAST0371
      IAE  = IELW + NEMAX                                                FAST0372
      IBE  = IAE  + NEMAX                                                FAST0373
      INSG = IBE  + NEMAX                                                FAST0374
      INPC = INSG + NVMAX                                                FAST0375
      IJSN = INPC + NVMAX*5                                              FAST0376
```

```
      IJSX = IJSN + NVMAX                                               FAST0377
      IXTR = IJSX + NVMAX                                               FAST0378
      ISUV = IXTR + NVMAX*3                                             FAST0379
      IATN = ISUV + NVMAX                                               FAST0380
      IATW = IATN + NIMAX                                               FAST0381
      IESB = IATW + NIMAX                                               FAST0382
      ISSH = IESB + NEMAX                                               FAST0383
      IDEN = ISSH + NEMOD                                               FAST0384
      INTG = IDEN + NNMAX                                               FAST0385
      IELF = INTG + NEMAX                                               FAST0386
      IFGW = IELF + NGMOD                                               FAST0387
      ITDS = IFGW + NGMAX                                               FAST0368
      IIDV = ITCS + NFMAX*3                                             FAST0388
      IIDR = IIDV + NVMOD                                               FAST0389
      IIDS = IIDR + NDMAX                                               FAST0390
      IVOL = IIDS + NDMAX                                               FAST0391
      ICDT = IVOL + NDMAX                                               FAST0392
      IXDT = ICDT + NDMAX*3                                             FAST0393
      IRSI = IXDT + NDMAX*3                                             FAST0394
      IALP = IRSI + NVMAX                                               FAST0395
      IVMD = IALP + NVMAX*5                                             FAST0396
      IGIM = IVMD + NVMAX*5                                             FAST0397
      IAIM = IGIM + NEMAX                                               FAST0398
      IALM = IAIM + NEMAX                                               FAST0399
      IALH = IALM + NEMAX                                               FAST0400
      IA   = IALH + NEMAX                                               FAST0401
      INS  = IA   + NSMAX*NAMAX                                         FAST0402
      IVEE = INS  + NRMAX*NBMAX                                         FAST0403
      IVAL = IVEE + NVMAX*NXMAX*5                                       FAST0404
      ISPW = IVAL + NVMAX*NXMAX*5                                       FAST0405
      ISPE = ISPW + NVMAX*NEMAX                                         FAST0406
      IATD = ISPE + NVMAX*NEMAX                                         FAST0407
      IRSP = IATD + NNMAX*NIMAX                                         FAST0408
      ISGT = IRSP + NFMAX*NGMOD                                         FAST0409
      IIDE = ISGT + NNMAX*NEMOD                                         FAST0410
      IIDI = IIDE + NNMAX*NORDER*NDOWN*NXSECT                           FAST0411
      ISGS = IIDI + NNMAX*INELAS*NXSECT                                 FAST0412
      NEXT = ISGS + NNMAX*NXSECT                                        FAST0413
      IFXA = NSTORE - NDMOD*NGMAX*MOMENT                                FAST0414
      IFXE = IFXA - NDMOD*NGMAX                                         FAST0415
      IFXT = IFXE - NDMOD*NGMAX                                         FAST0416
      IFXS = IFXT - NDMOD*NGMAX                                         FAST0417
      IFXP = IFXS - NDMOD*NGMAX                                         FAST0418
      INCP = IFXP - NSTMAX                                              FAST0419
      INRP = INCP - NSTMAX                                              FAST0420
      ISTP = INRP - NSTMAX                                              FAST0421
      INSC = ISTP - NSTMAX                                              FAST0422
```



```
      INRG = INSC - NSTMAX                                              FAST0423
      IST  = INRG - NSTMAX                                              FAST0424
      ISI  = IST  - NSMAX*2                                             FAST0425
      IW   = ISI  - NSMAX                                               FAST0426
      IV   = IW   - NSMAX                                               FAST0427
      IU   = IV   - NSMAX                                               FAST0428
      IND  = IU   - NSMAX                                               FAST0429
      IEC  = IND  - NEMAX                                               FAST0430
      IWC  = IEC  - NEMAX                                               FAST0431
      IES  = IWC  - NEMAX                                               FAST0432
      IWS  = IES  - NEMAX                                               FAST0433
      ISGR = IWS  - NEMAX                                               FAST0434
      IXMP = ISGR - NEMAX                                               FAST0435
      IF(KASE.EQ.1) GO TO 120                                           FAST0436
      DO 90 I=1,38                                                      FAST0437
      MOD = LCP(I+1) - LCP(I)                                           FAST0438
      MAX = LOC(I+1) - LOC(I)                                           FAST0439
      IF((MOD*MAX).EQ.0) GO TO 90                                       FAST0440
      IF(LOC(I).GT.LCP(I)) GO TO 80                                     FAST0441
      IF(LOC(I+1).GT.LCP(I+1)) GO TO 80                                 FAST0442
      J = LOC(I)                                                        FAST0443
      K = LCP(I)                                                        FAST0444
      CALL RESTOW(MAX,1,1,N(J),MOD,1,1,N(K))                            FAST0445
      GO TO 90                                                          FAST0446
   80 IF(LABEL(1).GE.0) WRITE(M2,2000)ARRAY(I)                          FAST0447
 2000 FORMAT(1X,39(1H*),A3,25H ARRAY CANNOT BE RESTORED,40(1H*))        FAST0448
      NERROR = NERROR + 1                                               FAST0449
   90 CONTINUE                                                          FAST0450
      DO 100 I=40,NARRAY                                                FAST0451
      IF(LCP(I-1).GE.LOC(I)) GO TO 100                                  FAST0452
      IF(LABEL(1).GE.0) WRITE(M2,2010)ARRAY(I-1)                        FAST0453
 2010 FORMAT(1X,35(1H*),A3,34H ARRAY MAY BE RESTORED INCORRECTLY,35(1H*)FAST0454
     1)                                                                 FAST0455
      NERROR = NERROR + 1                                               FAST0456
  100 CONTINUE                                                          FAST0457
      CALL RESTOW(NAMAX,NSMAX,1,N(IA),MAXA,MAXS,1,N(L1))                FAST0458
      CALL RESTOW(NBMAX,NRMAX,1,N(INS),MAXB,MAXR,1,N(L2))               FAST0459
      CALL RESTOW(NXMAX,5,NVMAX,N(IVEE),MAXX,5,MAXV,N(L3))              FAST0460
      CALL RESTOW(NXMAX,5,NVMAX,N(IVAL),MAXX,5,MAXV,N(L4))              FAST0461
      CALL RESTOW(NEMAX,NVMAX,1,N(ISPW),MAXE,MAXV,1,N(L5))              FAST0462
      CALL RESTOW(NEMAX,NVMAX,1,N(ISPE),MAXE,MAXV,1,N(L6))              FAST0463
      CALL RESTOW(NIMAX,NNMAX,1,N(IATD),MAXI,MAXN,1,N(L7))              FAST0464
      CALL RESTOW(NGMOD,NFMAX,1,N(IRSP),MODG,MAXF,1,N(L8))              FAST0465
      IF(NXSECT.NE.MODX) GO TO 120                                      FAST0466
      CALL RESTOW(NEMOD,NNMAX,1,N(ISGT),MODE,MAXN,1,N(L9))              FAST0467
      IF(NXSECT.EQ.0) GO TO 120                                         FAST0468
      CALL RESTOW(NDOWN,NORDER,NNMAX,N(IIDE),MODD,MODD,MAXN,N(L10))     FAST0469
```

```
      IF(INELAS.LE.0) GO TO 110                                         FAST0470
      CALL RESTOW(INELAS,NMMAX,1,N(IDI),MODI,MAXM,1,N(L11))             FAST0471
  110 CALL RESTOW(NNMAX,1,1,N(ISGS),MAXN,1,1,N(L12))                    FAST0472
  120 IF(IN11.LE.0) GO TO 130                                           FAST0473
      IF(LABEL(15).GE.0) WRITE(M2,2030)(ARRAY(I),LOC(I),I=1,73)         FAST0474
  130 CONTINUE                                                          FAST0475
 2030 FORMAT(1X,43(1H*),22HDATA ARRAY STORAGE MAP,43(1H*)/6(4X,14HARRAY FAST0476
     1LOCATION)/(6(6X,A3,I9)))                                          FAST0477
      RETURN                                                            FAST0478
      END                                                               FAST0479
$IBFTC RESTOR  M94/2,XR7                                                FAST0480
CRESTOW*RESTORE MULTIDIMENSIONAL ARRAYS WITH NEW DIMENSIONS*T. M. JORDANFAST0481
      SUBROUTINE RESTOW(IA,JA,KA,NA,IB,JB,KB,NB)                        FAST0482
      DIMENSION NA(IA,JA,KA),NB(IB,JB,KB)                               FAST0483
      DO 20 K=1,KA                                                      FAST0484
      DO 20 J=1,JA                                                      FAST0485
      DO 10 I=1,IA                                                      FAST0486
      NC = 0                                                            FAST0487
      IF(I.GT.IB) GO TO 10                                              FAST0488
      IF(J.GT.JB) GO TO 10                                              FAST0489
      IF(K.GT.KB) GO TO 10                                              FAST0490
      NC = NB(I,J,K)                                                    FAST0491
   10 NA(I,J,K) = NC                                                    FAST0492
   20 CONTINUE                                                          FAST0493
      RETURN                                                            FAST0494
      END                                                               FAST0495
$IBFTC GEOM     M94/2,XR7                                               FAST0496
CGEOMIN*SURFACE AND REGION INPUT FOR PROGRAM FASTER*T.M.JORDAN*WANL*66  FAST0497
      SUBROUTINE GEOMIN(L1,L2,L3,NTP,AZ,A,ISV,MTL,NS,RHO,XR)            FAST0498
      DIMENSION AA(7),ADM(9),IDM(12)                                    FAST0499
      EQUIVALENCE (AA(1),ADM(1))                                        FAST0500
      DIMENSION NTP(1),AZ(1),A(L1,L3),ISV(1),MTL(1),NS(L2,L3),RHO(1),   FAST0501
     1          XR(3,1)                                                 FAST0502
      COMMON/TAPEIO/M1     ,M2                                          FAST0503
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST0504
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST0505
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST0506
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST0507
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST0508
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST0509
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST0510
      COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)    FAST0511
C QUADRIC SURFACES                                                      FAST0512
      IF(IN2.LE.0)GO TO 300                                             FAST0513
      DO 290 N=1,IN2                                                    FAST0514
      CALL READS(IDM,AA)                                                FAST0515
      I = IDM(1)                                                        FAST0516
```

```
      NTP(I) = IDM(2)                                                   FAST0517
      NEX = IDM(3)                                                      FAST0518
      IF(NEX.GT.0)GO TO 230                                             FAST0519
C EXPANDED FORM                                                         FAST0520
      AZ(I) = AA(1)                                                     FAST0521
      MAX = MINO(NBMAX,6)                                               FAST0522
      DO 220 J=1,MAX                                                    FAST0523
  220 A(J,I) = AA(J+1)                                                  FAST0524
      IF(NTP(I).GT.6)CALL READE(NTP(I)-6,A(7,I))                        FAST0525
      GO TO 290                                                         FAST0526
  230 NGT = (NEX+2)/3                                                   FAST0527
      NEX = NEX - 3*(NGT-1)                                             FAST0528
      DO 231 J=1,NBMAX                                                  FAST0529
  231 A(J,I) = 0.0                                                      FAST0530
      AZ(I) = 0.0                                                       FAST0531
      GO TO(232,233,235,260,255),NGT                                    FAST0532
  232 NTP(I) = NEX                                                      FAST0533
      AZ(I) = -AA(1)                                                    FAST0534
      A(NEX,I) = 1.0                                                    FAST0535
      GO TO 290                                                         FAST0536
  233 FST = AA(3) - AA(1)                                               FAST0537
      D = -1.0                                                          FAST0538
      K = 0                                                             FAST0539
      DO 234 J=1,3                                                      FAST0540
      IF(J.EQ.NEX)GO TO 234                                             FAST0541
      D = -D                                                            FAST0542
      K = K + 1                                                         FAST0543
      NTP(I) = J                                                        FAST0544
      A(J,I) = D*FST/(AA(K+2) - AA(K))                                  FAST0545
      AZ(I) = AZ(I) - AA(K)*A(J,I)                                      FAST0546
  234 CONTINUE                                                          FAST0547
      GO TO 290                                                         FAST0548
C CONE PARALLEL TO NEX-AXIS                                             FAST0549
  235 CONTINUE                                                          FAST0550
      XX = (AA(6) - AA(4))/(AA(5) - AA(3))                              FAST0551
      YY = XX*AA(3) - AA(4)                                             FAST0552
      NTP(I) = 6                                                        FAST0553
      XX = XX**2                                                        FAST0554
      AZ(I) = YY**2 - XX*(AA(1)**2 + AA(2)**2)                          FAST0555
      K = 0                                                             FAST0556
      DO 250 J=1,3                                                      FAST0557
      IF(J.NE.NEX)GO TO 240                                             FAST0558
      A(J,I) = 2.0*YY                                                   FAST0559
      A(J+3,I) = 1.0                                                    FAST0560
      GO TO 250                                                         FAST0561
  240 K = K + 1                                                         FAST0562
      A(J,I) = 2.0*XX*AA(K)                                             FAST0563
```

```
      A(J+3,I) = -XX                                                    FAST0564
  250 CONTINUE                                                          FAST0565
      GO TO 290                                                         FAST0566
C ELLIPTICAL CYLINDER AND ELLIPSOID                                     FAST0567
  255 NEX = 0                                                           FAST0568
  260 CONTINUE                                                          FAST0569
      AZ(I) = -1.0                                                      FAST0570
      K = -1                                                            FAST0571
      DO 280 J=1,3                                                      FAST0572
      IF(J.EQ.NEX)GO TO 280                                             FAST0573
  270 NTP(I) = J + 3                                                    FAST0574
      K = K + 2                                                         FAST0575
      A(J+3,I) = 1.0/AA(K+1)**2                                         FAST0576
      A(J,I) = -2.0*AA(K)*A(J+3,I)                                      FAST0577
      AZ(I) = AZ(I) + AA(K)**2*A(J+3,I)                                 FAST0578
  280 CONTINUE                                                          FAST0579
  290 CONTINUE                                                          FAST0580
C REGION DEFINITIONS                                                    FAST0581
  300 IF(IN3.LE.0)GO TO 330                                             FAST0582
      FST = 1.0                                                         FAST0583
      IF(NUNITD.EQ.1)FST = 0.6025/1.C0797                               FAST0584
      DO 320 N=1,IN3                                                    FAST0585
      CALL READR(IDM,ADM)                                               FAST0586
      I = IDM(1)                                                        FAST0587
      ISV(I) = IDM(2)                                                   FAST0588
      MTL(I) = IDM(3)                                                   FAST0589
      RHO(I) = FST*ADM(1)                                               FAST0590
      DO 305 J=1,3                                                      FAST0591
  305 XR(J,I) = ADM(J+1)                                                FAST0592
      DO 312 J=1,NBMAX                                                  FAST0593
  312 NS(J,I) = 0                                                       FAST0594
      MOD = MIN0(NBMAX,9)                                               FAST0595
      DO 315 J=1,MOD                                                    FAST0596
  315 NS(J,I) = 1000*IDM(J+3)                                           FAST0597
      IF(NBMAX.LE.9) GO TO 320                                          FAST0598
      IF(IDM(12).GT.-1) GO TO 320                                       FAST0599
      CALL READI(NBMAX-8,NS(9,I))                                       FAST0600
      DO 318 J=9,NBMAX                                                  FAST0601
  318 NS(J,I) = 1000*NS(J,I)                                            FAST0602
  320 CONTINUE                                                          FAST0603
  330 IF(IN4.LE.0)GO TO 340                                             FAST0604
      DO 334 I=1,NRMAX                                                  FAST0605
      DO 331 J=1,3                                                      FAST0606
      ADM(J) = XR(J,I)                                                  FAST0607
      ADM(J+3) = ADM(J)**2                                              FAST0608
      K = J + 1 - 3*(J/3)                                               FAST0609
  331 ADM(J+6) = ADM(J)*XR(K,I)                                         FAST0610
```

196

```
      DO 333 J=1,NBMAX                                                  FAST0611
      IF(NS(J,I).EQ.0)GO TO 334                                         FAST0612
      K = NS(J,I)/1000                                                  FAST0613
      K = K - 1000*(K/1000)                                             FAST0614
      MAX = NTP(K)                                                      FAST0615
      NS(J,I) = 1000*K                                                  FAST0616
      FST = AZ(K)                                                       FAST0617
      DO 332 L=1,MAX                                                    FAST0618
  332 FST = FST + ADM(L)*A(L,K)                                         FAST0619
      IF(FST.LT.0.0)NS(J,I) = NS(J,I) + 1000000                         FAST0620
  333 CONTINUE                                                          FAST0621
  334 CONTINUE                                                          FAST0622
  340 IF(IN5.LE.0)GO TO 370                                             FAST0623
      DO 360 I=1,NRMAX                                                  FAST0624
      NGS = I                                                           FAST0625
  350 NGS = LOCATE(NGS+1,XR(1,I))                                       FAST0626
      IF(NGS.EQ.I)GO TO 360                                             FAST0627
      IF(LABEL(1).GE.0)WRITE(M2,2000)I,NGS                              FAST0628
 2000 FORMAT(1X,22(1H*),35HGEOMETRY ERROR, THE POINT IN REGION,I5,18H ISFAST0629
     1 ALSO IN REGION,I5,22(1H*))                                       FAST0630
      NERROR = NERROR + 1                                               FAST0631
      IF(NGS.GT.0)GO TO 350                                             FAST0632
  360 CONTINUE                                                          FAST0633
  370 RETURN                                                            FAST0634
      END                                                               FAST0635
$IBFTC SOURSE  M94/2,XR7                                                FAST0636
CSOURCE*FIXED SOURCE INPUT AND NORMALIZATION FOR PROGRAM FASTER*T.JORDANFAST0637
      SUBROUTINE SOURCE(L1,L2,L3,L4,ELL,ELW,AE,BE,NSG,NPC,JSN,JSX,SUV,  FAST0638
     1                 XTR,VEE,VAL,SPW,SPE,ISV,E,EN,EBG,ENG)            FAST0639
      DIMENSION IDM(12),ADM(4)                                          FAST0640
      DIMENSION ELL(1),ELW(1),AE(1),BE(1),NSG(1),NPC(5,1),JSN(1),JSX(1),FAST0641
     1          SUV(1),XTR(3,1),VEE(L2,L3,L4),VAL(L2,L3,L4),SPW(L1,L4), FAST0642
     2          SPE(L1,L4),ISV(1),E(1),EN(1),EBG(1),ENG(1)              FAST0643
      COMMON/TAPEID/M1     ,M2                                          FAST0644
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST0645
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST0646
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NOOWN ,INELAS,NTRANS,   FAST0647
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST0648
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST0649
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST0650
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST0651
      COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)    FAST0652
      IF(IN2.LE.0)GO TO 330                                             FAST0653
      CALL READE(NEMAX+1,ELL)                                           FAST0654
      DO 10 I=1,NEMAX                                                   FAST0655
      ELW(I) = ELL(I) - ELL(I+1)                                        FAST0656
      AE(I) = 1.0                                                       FAST0657
```

197

Astronuclear Laboratory

```
   10 BE(I) = 0.0
C TOTAL SOURCE, ERROR CRITERION, TRANSLATION VECTOR
  330 IF(IN3.LE.0)GO TO 505
      DO 430 NNN=1,IN3
      CALL READR(IDM,ADM)
      VOLUME = 1.0
      I = IDM(1)
      NSG(I) = IDM(2)
      N = NSG(I)
      DO 405 J=1,5
      MAX = IDM(J+2)
      IF(MAX.GT.0) GO TO 299
      K = -MAX
      MAX = NPC(J,K)
      DO 200 L=1,MAX
      VEE(L,J,I) = VEE(L,J,K)
  200 VAL(L,J,I) = VAL(L,J,K)
      NPC(J,I) = MAX
      IF(MAX - 1) 405,405,210
  299 NPC(J,I) = MAX
      CALL READEE(MAX,VEE(1,J,I),VAL(1,J,I))
      IF(MAX.LE.1)GO TO 405
      FST = 0.0
C     DO 987 K=1,MAX
C     IF(VEE(K,J,I).EQ.0.0) VEE(K,J,I) = 1.0E-30
C 987 CONTINUE
      DO 395 K=2,MAX
      BB = 0.0
      AB = 0.0
      IF(VEE(K,J,I).EQ.VEE(K-1,J,I)) GO TO 395
      BB = (VAL(K,J,I) - VAL(K-1,J,I))/(VEE(K,J,I) - VEE(K-1,J,I))
      AB = VAL(K,J,I) - BB*VEE(K,J,I)
  395 FST = FST+AB*(VEE(K,J,I)**(N+1) - VEE(K-1,J,I)**(N+1))/FLOAT(N+1)
     1          +BB*(VEE(K,J,I)**(N+2) - VEE(K-1,J,I)**(N+2))/FLOAT(N+2)
      DO 400 K=1,MAX
  400 VAL(K,J,I) = VAL(K,J,I)/FST
  210 CONTINUE
      IF(J.LE.3)VOLUME = VOLUME*(VEE(MAX,J,I)**(N+1)-VEE(1,J,I)**(N+1))
  405 N = 0
      ISP = IDM(10)
      MAX = IDM(8)
      IF(MAX.GT.0)GO TO 298
      K = -MAX
      JSN(I) = JSN(K)
      JSX(I) = JSX(K)
      FST = 0.0
      EST = 0.0
```



```
      DO 220 J=1,NEMAX
      SPW(J,I) = SPW(J,K)
      SPE(J,I) = SPE(J,K)
      FST = FST + SPW(J,I)
  220 EST = EST + SPW(J,I)*SPE(J,I)
      GO TO 297
  298 CONTINUE
      IF(ISP.GT.1) GO TO 100
C INPUT SPECTRUM IS DIFFERENTIAL IN PARTICLES OR INTENSITY
      IF(IDM(11).GT.0) GO TO 600
      CALL READEE(MAX,E,EN)
      GO TO 601
  600 IF(IDM(11).EQ.2) CALL READE(MAX,E)
      CALL READE(MAX,EN)
  601 CONTINUE
      IF(ISP.LE.0)GO TO 354
C DIFFERENTIAL IN INTENSITY
      DO 350 J=1,MAX
  350 EN(J) = EN(J)/E(J)
      GO TO 354
C INPUT SPECTRUM IS GROUPWISE INTEGRATED, MAX = NO. GROUPS
  100 IF(IDM(11).EQ.0) GO TO 351
      IF(IDM(11).EQ.2) CALL READE(MAX+1,EBG)
      CALL READE(MAX,ENG)
      GO TO 110
  351 CALL READEE(MAX+1,EBG,ENG)
  110 CONTINUE
      IF(ISP.EQ.2)GO TO 355
C TOTAL MEV IN GROUP, DIVIDE BY AVERAGE ENERGY
      DO 352 J=1,MAX
  352 ENG(J) = 2.0*ENG(J)/(EBG(J) + EBG(J+1))
C TOTAL PARTICLES IN GROUP, DIVIDE BY GROUP WIDTH
  355 K = 0
      DO 353 J=1,MAX
      FST = ENG(J)/(EBG(J) - EBG(J+1))
      DO 353 L=1,2
      K = K + 1
      M = J + L - 1
      E(K) = EBG(M)
  353 EN(K) = FST
      MAX = K
  354 FST = 0.0
      EST = 0.0
      JSN(I) = NEMAX
      JSX(I) = 1
      DO 415 J=1,NEMAX
      SPW(J,I) = 0.0
```

Astronuclear Laboratory

200

```
      SPE(J,I) = 0.0                                                    FAST0752
      DO 410 K=2,MAX                                                    FAST0753
      EMAX = AMIN1(ELL(J),E(K-1))                                       FAST0754
      EMIN = AMAX1(ELL(J+1),E(K))                                       FAST0755
      IF(EMIN.GE.EMAX)GO TO 410                                         FAST0756
      BB = (EN(K-1) - EN(K))/(E(K-1) - E(K))                            FAST0757
      AA = EN(K) - BB*E(K)                                              FAST0758
      DEL = (EMAX**2 - EMIN**2)/2.0                                     FAST0759
      SPW(J,I) = SPW(J,I) + AA*(EMAX - EMIN) + BB*DEL                   FAST0760
      SPE(J,I) = SPE(J,I) + AA*DEL + BB*(EMAX**3 - EMIN**3)/3.0         FAST0761
  410 CONTINUE                                                          FAST0762
      IF(SPW(J,I).EQ.0.0)GO TO 415                                      FAST0763
      JSN(I) = MIN0(JSN(I),J)                                           FAST0764
      JSX(I) = MAX0(JSX(I),J)                                           FAST0765
      FST = FST + SPW(J,I)                                              FAST0766
      EST = EST + SPE(J,I)                                              FAST0767
      SPE(J,I) =  SPE(J,I)/SPW(J,I)                                     FAST0768
  415 CONTINUE                                                          FAST0769
  797 CONTINUE                                                          FAST0770
      DO 420 J=1,3                                                      FAST0771
  420 XTR(J,I) = ADM(J+1)                                               FAST0772
      EAVE = EST/FST                                                    FAST0773
      IF(IDM(9) - 1) 421,422,423                                        FAST0774
C NORMALIZE TO TOTAL PARTICLES                                          FAST0775
  421 FST = ADM(1)/FST                                                  FAST0776
      SUV(I) = ADM(1)*EAVE                                              FAST0777
      GO TO 424                                                         FAST0778
C NORMALIZE TO TOTAL MEV                                                FAST0779
  422 FST = ADM(1)/EST                                                  FAST0780
      SUV(I) = ADM(1)                                                   FAST0781
      GO TO 424                                                         FAST0782
C SCALE INPUT SPECTRUM                                                  FAST0783
  423 FST = ADM(1)                                                      FAST0784
      SUV(I) = ADM(1)*EST                                               FAST0785
  424 CONTINUE                                                          FAST0786
      DO 425 J=1,NEMAX                                                  FAST0787
  425 SPW(J,I) = FST*SPW(J,I)                                           FAST0788
  430 CONTINUE                                                          FAST0789
  505 IF(IN4.GT.0)CALL READIS(IN4,ISV)                                  FAST0790
      DO 510 I=1,NRMAX                                                  FAST0791
      IF(ISV(I).GT.NVMAX)ISV(I) = 0                                     FAST0792
  510 CONTINUE                                                          FAST0793
      RETURN                                                            FAST0794
      END                                                               FAST0795
$IBFTC XSECT    M94/2,XR7                                               FAST0796
CINSECT*CROSS SECTION INPUT AND MIXING FOR PROGRAM FASTER*T.M.JORDAN*66*FAST0797
      SUBROUTINE INSECT(L1,L2,L3,L4,L5,L6,ESB,DEN,SSH,SGT,IDE,IDI,      FAST0798
```

Astronuclear Laboratory

201

```
     1SGS,MTL,RHO,XST,ATD,JMX,XSE,XSI,EAC,ELL,L7)                       FAST0799
      DIMENSION ADM(8)                                                  FAST0800
      EQUIVALENCE (ADM(1),ATW),(ADM(2),ZEE)                             FAST0801
      DIMENSION ESB(1),DEN(1),SSH(1),SGT(L1,L6),IDE(L2,L3,L6),          FAST0802
     1          IDI(L4,L6)        ,SGS(1 ,1 ),MTL(1),RHO(1),XST(1),     FAST0803
     2          ATD(1),JMX(1),XSE(L5,L2,L6),XSI(L7,L6)                  FAST0804
      DIMENSION EAC(L1,L6),ELL(1)                                       FAST0805
      COMMON/TAPEID/M1     ,M2                                          FAST0806
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST0807
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST0808
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST0809
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST0810
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST0811
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST0812
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST0813
      COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)    FAST0814
      DIMENSION XSET(2)                                                 FAST0815
      EQUIVALENCE (XSET(1),XMB),(XSET(2),FST)                           FAST0816
      IF(IN2.GT.0)CALL READE(NEMAX,ESB)                                 FAST0817
      IF(IN3.LE.0)GO TO 210                                             FAST0818
      NERRX = 0                                                         FAST0819
      NEMIN = NEMOD - NXSECT                                            FAST0820
      DO 10 I=1,NMMAX                                                   FAST0821
      DEN(I) = 0.0                                                      FAST0822
      DO 10 J=1,NEMIN                                                   FAST0823
   10 SGT(J,I) = 0.0                                                    FAST0824
      IF(NXSECT.NE.1)GO TO 50                                           FAST0825
      DO 30 I=1,NMMAX                                                   FAST0826
      DO 20 J=1,NORDER                                                  FAST0827
      DO 20 K=1,NDOWN                                                   FAST0828
   20 IDE(K,J,I) = 0                                                    FAST0829
   30 CONTINUE                                                          FAST0830
      IF(INELAS.GT.0) CALL ZEROUT(INELAS*NMMAX,IDI)                     FAST0831
      CALL ZEROUT(IN1,SGS)                                              FAST0832
   50 NN = 0                                                            FAST0833
      NNMOD = IN1 + 1                                                   FAST0834
      CALL ZEROUT(NEMOD*(NIMAX+NMMAX),EAC)                              FAST0835
      NN = 1                                                            FAST0836
      DO 200 I=1,NIMAX                                                  FAST0837
      MOD = MIN0(8,NMMAX+2)                                             FAST0838
      CALL READE(MOD,ADM)                                               FAST0839
      MOD = MOD - 2                                                     FAST0840
      DO 65 J=1,MOD                                                     FAST0841
   65 ATD(J) = ADM(J+2)                                                 FAST0842
      IF(NMMAX.GT.6)CALL READE(NMMAX-6,ATD(7))                          FAST0843
      IF(NUNITD.EQ.0)GO TO 85                                           FAST0844
      FST = 0.6025/ATW                                                  FAST0845
```

```
      DO 75 J=1,NMMAX
   75 ATD(J) = FST*ATD(J)
   85 CALL READE(NEMIN,XST)
      IF(NUNITX.EQ.0)GO TO 105
      FST = ATW/0.6025
      DO 95 J=1,NEMIN
   95 XST(J) = FST*XST(J)
  105 IF(NXSECT.GT.0) GO TO 105
COMPTON ENERGY ABSORBTION COEFFICIENT
      DO 106 J=1,NEMOD
      XMA = ELL(J)/0.511
      XMB = 1.0 + 2.0*XMA
  106 EAC(J,I)=ELL(J)*(XST(J)-0.249375*ZEE*(ALOG(XMB)/XMA**3+2.0*(1.0+
     1XMA)*(2.0*XMA**2-XMB)/(XMA*XMB)**2+8.0*XMA**2/(3.0*XMB**3)))
  109 IF(I.GT.1) GO TO 125
      DO 115 J=1,NEMIN
  115 SSH(J) = XST(J)
      IF(NXSECT.EQ.0) GO TO 100
      DO 116 J=1,NEMOD
      EAC(J,I) = 0.0
      IF(J.GT.1) EAC(J,I) = SSH(J-1)
      IF(J.LT.NEMOD) EAC(J,I) = EAC(J,I) + SSH(J)
  116 EAC(J,I) = ELL(J)*EAC(J,I)/4.0
      GO TO 398
  125 IF(NXSECT.EQ.0)GO TO 100
      MOD = MIN0(24,INELAS+3)
      CALL READI(MOD,JMX)
      NSIGT = 1
      IF(JMX(1).GT.0) NSIGT = 0
      LMAX = IABS(JMX(1))
      NDSM = JMX(2)
      KMAX = JMX(3)
      IF(KMAX.GT.21)CALL READI(KMAX-21,JMX(25))
      DO 80 L=1,LMAX
      DO 80 K=1,NDSM
   80 CALL READE(NEMAX+1-K,XSE(1,K,L))
C TRANSPORT CORRECTION OF P-ZERO IN GROUP FROM P-ONE SCATTER
      XMA = ATW/(1.0*(ATW + 1.0)**2)
      IF(NUNITX.GT.0) XMA = XMA*ATW/0.6025
      LMOD = MIN0(2,LMAX)
      DO 82 J=1,NEMAX
      FST = 0.0
      XMB = 0.0
      NDSMOD = MIN0(NDSM,NEMAX+1-J)
      DO 81 L=1,LMOD
      DO 81 K=1,NDSMOD
   81 XSET(L) = XSET(L) + XSE(J,K,L)
```



```
      XMUL = FST/(3.0*XMB)
      XMUC = XMUL**2 - 1.0
      XMUC = (XMUC + XMUL*SQRT(ATW**2 + XMUC))/ATW
      EAC(J,I) = XMA*XMB*(1.0 - XMUC)
      IF(LMAX.GT.NORDER) XSE(J,1,1) = XSE(J,1,1) - FST/3.0
      IF(NUNITX.GT.0) FST = FST*ATW/0.6025
   82 XST(J) = XST(J) + FST*FLOAT(NSIGT - 1/NORDER)/3.0
   83 CONTINUE
      LMAX = MIN0(NORDER,LMAX)
      XMA = 0.0
      DO 397 J=1,NEMOD
      XMB = 0.0
      IF(J.LT.NEMOD) XMB = EAC(J,I)
      EAC(J,I) = ELL(J)*(XMA + XMB)
  397 XMA = XMB
      IF(KMAX.LE.0)GO TO 398
C INELASTIC TRANSFER
      DO 90 K=1,KMAX
   90 CALL READE(JMX(K+3),XSI(1,K))
  398 DO 117 J=1,NEMOD,NEMAX
  117 EAC(J,I) = 2.0*EAC(J,I)
C COMBINE MATERIAL TOTALS
  100 DO 190 M=1,NMMAX
      IF(ATD(M).EQ.0.0)GO TO 190
      DO 110 J=1,NEMIN
  110 SGT(J,M) = SGT(J,M) + ATD(M)*XST(J)
      L = NIMAX + M
      DO 111 J=1,NEMOD
  111 EAC(J,L) = EAC(J,L) + ATD(M)*EAC(J,I)
      IF(NXSECT.GT.0)GO TO 120
      DEN(M) = DEN(M) + 0.49875*ATD(M)*ZEE
      GO TO 190
  120 IF(I.GT.1)GO TO 130
      DEN(M) = ATD(M)
      GO TO 190
  130 GST = ATD(M)
      IF(NUNITX.NE.0)GST = GST*ATW/0.6025
      DO 165 L=1,LMAX
      DO 165 K=1,NDSM
      MM = IDE(K,L,M)
      MAX = NEMOD - K
      IF (MM.GT.0) GO TO 140
      MM = NN
      NN = NN + MAX
      IDE(K,L,M) = MM
  140 IF((MM + MAX).LE.NNMOD) GO TO 150
      NERROR = NERROR + 1
```

```
      GO TO 165                                                         FAST0940
  150 DO 160 J=1,MAX                                                    FAST0941
  160 SGS(J,MM) = SGS(J,MM) + GST*XSE(J,K,L)                            FAST0942
  165 CONTINUE                                                          FAST0943
      IF(KMAX.LE.0)GO TC 190                                            FAST0944
C INELASTIC NEXT                                                        FAST0945
      DO 185 K=1,KMAX                                                   FAST0946
      MM1= IDI(K,M)                                                     FAST0947
      MAX = JMX(K+3)                                                    FAST0948
      IF(MM1.GT.0) GO TO 170                                            FAST0949
      MM = NN                                                           FAST0950
      NN = NN + MINO(NTRANS,NEMOD-K)                                    FAST0951
      MM1 = MM                                                          FAST0952
      GO TO 173                                                         FAST0953
  170 MM  = MM1/1000                                                    FAST0954
  173 IDI(K,M) = 1000*MM  + MAXO(MAX,MM1-1000*MM)                       FAST0955
      IF((MM + MAX).LE.NNMCO) GO TO 175                                 FAST0956
      NERROR = NERROR + 1                                               FAST0957
      GO TO 185                                                         FAST0958
  175 DO 180 J=1,MAX                                                    FAST0959
  180 SGS(J,MM) = SGS(J,MM) + GST*XSI(J,K)                              FAST0960
  185 CONTINUF                                                          FAST0961
  190 CONTINUE                                                          FAST0962
  200 CONTINUF                                                          FAST0963
  390 CONTINUE                                                          FAST0964
      NNMAX = NN - 1                                                    FAST0965
      IF(LABEL(1).GE.0) WRITE(M2,2000)NNMAX,IN1                         FAST0966
 2000 FORMAT(1X,19(1H*),34HSCATTERING CROSS SECTIONS REQUIRED,I6,3H OF, FAST0967
     1I6,20H AVAILABLE LOCATIONS,19(1H*))                               FAST0968
      IF(IN6.LE.0) GO TC 210                                            FAST0969
      DO 510 I=1,NMMAX                                                  FAST0970
      DO 500 J=1,NEMIN,8                                                FAST0971
      MAX = MINO(J + 7,NEMIN)                                           FAST0972
      IF(LABEL(1).GE.0) WRITE(M2,3000)I,(SGT(K,I),K=J,MAX)              FAST0973
 3000 FORMAT(1X,6HSIGMAT,I5,1P8E12.4)                                   FAST0974
  500 CONTINUE                                                          FAST0975
  510 CONTINUE                                                          FAST0976
      MAXR = NIMAX + NMMAX                                              FAST0977
      DO 530 I=1,MAXR                                                   FAST0978
      K = I                                                             FAST0979
      IF(K.GT.NIMAX) K = K - NIMAX                                      FAST0980
      DO 520 J=1,NEMOD,8                                                FAST0981
      MAX = MINO(J + 7,NEMOD)                                           FAST0982
      IF(LABEL(1).GE.0) WRITE(M2,3010)K,(EAC(L,I),L=J,MAX)              FAST0983
 3010 FORMAT(1X,6HE-DUMP,I5,1P8E12.4)                                   FAST0984
  520 CONTINUE                                                          FAST0985
  530 CONTINUE                                                          FAST0986
```



```
  210 IF(IN4.GT.0)CALL READIS(IN4,MTL)                                  FAST0987
      IF(IN5.LE.0)GO TO 250                                             FAST0988
      IF(NUNITO.EQ.0)GO TO 230                                          FAST0989
      FST = 0.6025/1.00797                                              FAST0990
      DO 220 I=1,NRMAX                                                  FAST0991
  220 RHO(I) = RHO(I)/FST                                               FAST0992
  230 CALL READES(IN5,RHO)                                              FAST0993
      IF(NUNITO.EC.0)GO TO 250                                          FAST0994
      DO 240 I=1,NRMAX                                                  FAST0995
  240 RHO(I) = FST*RHO(I)                                               FAST0996
  250 RETURN                                                            FAST0997
      END                                                               FAST0998
$IBFTC ASKFOR  M94/2,XR7                                                FAST0999
CRESULT*RESPONSE FUNCTION AND DETECTOR INPUT FOR FASTER*T.M.JORDAN*WANL*FAST1000
      SUBROUTINE RESULT(L1,L2,NTG,ELF,FGW,TDS,RSP,IDR,IDS,VOL,CDT,XDT,  FAST1001
     1ELL,MTL,RHO,EAC,L3)                                               FAST1002
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST1003
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST1004
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST1005
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1006
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1007
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1008
     6              NPRINT,NUNITS,NUMBER,KALTDE                         FAST1009
      COMMON H(1)                                                       FAST1010
      COMMON/REGSCA/NSRMAX,INSR                                         FAST1011
      COMMON/INDEXS/MDUM(25),IIDV,MDUM(47)                              FAST1012
      DIMENSION MTL(1),RHO(1),EAC(L3,L2)                                FAST1013
      DIMENSION NTG(1),ELF(1),FGW(1),TDS(3,1),RSP(L1,L2),IDR(1),IDS(1), FAST1014
     1          VOL(1),CDT(3,1),XDT(3,1),ELL(1)                         FAST1015
      DIMENSION ADM(7),BDM(3),IDM(3)                                    FAST1016
      COMMON/TAPEID/M1    ,M2                                           FAST1017
      COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)    FAST1018
      IF(NGMAX.LT.NEMAX)GO TO 20                                        FAST1019
      DO 10 I=1,NEMAX                                                   FAST1020
   10 NTG(I) = 1                                                        FAST1021
   20 IF(IN2.GT.0) CALL READI(NEMAX,NTG)                                FAST1022
   30 ELF(1) = ELL(1)                                                   FAST1023
      DO 40 I=1,NEMAX                                                   FAST1024
      J = NTG(I)                                                        FAST1025
   40 ELF(J+1) = ELL(I+1)                                               FAST1026
      DO 50 I=1,NGMAX                                                   FAST1027
   50 FGW(I) = ELF(I) - ELF(I+1)                                        FAST1028
   60 IF(IN3.LE.0)GO TO 120                                             FAST1029
      DO 110 N=1,IN3                                                    FAST1030
      CALL READF(IDM,BDM,ADM)                                           FAST1031
      I = IDM(1)                                                        FAST1032
      DO 70 J=1,3
```

```
   70 TOS(J,I) = BDM(J)
      IF(IDM(2).GE.0) GC TO 75
      J = -IDM(2)
      M = MTL(J) + NIMAX
      RSP(1,I) = RHO(J)*EAC(1,1)
      IF(M.GT.NIMAX) RSP(1,I) = RSP(1,I) + EAC(1,M)
      DO 11 K=1,NEMAX
      L = NTG(K)
      RSP(L+1,I) = RHO(J)*EAC(K+1,1)
      IF(M.GT.NIMAX) RSP(L+1,I) = RSP(L+1,I) + EAC(K+1,M)
   11 CONTINUE
      IF(LABEL(1+NGMOD/7).GE.0) WRITE(M2,2000)(RSP(J,I),J=1,NGMOD)
 2000 FORMAT(1X,11HENERGY DUMP,1P8E12.4)
      GO TO 95
   75 CONTINUE
      MAX = MINO(5,NGMOD)
      DO 80 J=1,MAX
   80 RSP(J,I) = ADM(J+1)
      IF(NGMOD.GT.5)CALL READE(NGMOD-5,RSP(6,I))
      IF(IDM(2).EQ.0)GO TO 100
      DO 90 J=1,NGMOD
   90 RSP(J,I) = ELF(J)*RSP(J,I)
   95 CONTINUE
  100 DO 110 J=1,NGMOD
  110 RSP(J,I) = ADM(1)*RSP(J,I)
  120 IF(IN4.LE.0)GO TO 552
      DO 130 I=1,3
  130 BDM(I) = 0.0
      DO 150 N=1,IN4
      CALL READS(IDM,ADM)
      I = IDM(1)
      IDR(I) = IDM(2)
      IDS(I) = IDM(3)
      VOL(I) = ADM(1)
      DO 140 J=1,3
      CDT(J,I) = ADM(J+1)
  140 XDT(J,I) = ADM(J+4)
  150 FST = VECTOR(BDM,CDT(1,I),CDT(1,I))
  552 NPDMAX = 0
      NVDMAX = 0
      DO 556 I=1,NDMAX
      IF(IDR(I).GT.0)GO TO 554
      NPDMAX = NPDMAX + 1
      GO TO 556
  554 IF(IDS(I).LE.0)NVDMAX = NVDMAX + 1
  556 CONTINUE
      NVDMOD = NDMAX - NPDMAX
```



```
      NSDMAX = NVDMOD - NVDMAX
      NPDMOD = NPDMAX
      IF(IN5.GT.0) CALL READI(NVMOD,H(IIDV))
      IF(IN6.GT.0) CALL READI(NSRMAX,H(1))
      RETURN
      END
$IBFTC RANDMN  M94/2,XR7
CRANDOM*SAMPLING PARAMETER INPUTS FOR PROGRAM FASTER*T.M.JORDAN*WANL*66*
      SUBROUTINE RANDOM(RSI,ALP,VMD,GIM,AIM,ALM,ALH,NPC,VEE,L1,L2)
      DIMENSION NPC(5,1),VEE(L1,L2,L1),RAT(5)
      DIMENSION RSI(1),ALP(5,1),VMD(5,1),GIM(1),AIM(1),ALM(1),ALH(1)
      COMMON/TAPEID/M1    ,M2
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA   ,ATB   ,ATC   ,
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,
     2              KMIN  ,KMAX  ,JBAR  ,NZERO
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,
     5              NVDMAX,NVDMOD,NPCINT,MODELP,MODELQ,MODELU,MODELV,
     6              NPRINT,NUNITS,NUMBER,KALIDE
      COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)
      IF(IN2.GT.0)CALL READE(8,RADIUS)
      IF(IN3.GT.0)CALL READE(NVMAX,RSI)
      IF(MODELP.GT.0)GO TO 30
      IF(NVMAX .EQ.1) GO TO 30
      FST = 0.0
      DO 10 I=1,NVMAX
   10 FST = FST + RSI(I)
      DO 20 I=1,NVMAX
   20 RSI(I) = RSI(I)/FST
   30 IF(IN4.LE.0)GO TO 793
      DO 560 I=1,NVMAX
      CALL READE(5,VMD(1,I))
      CALL READE(5,RAT)
      DO 40 J=1,5
      ALP(J,I) = 0.0
      K = NPC(J,I)
      IF(K.EQ.1)GO TO 40
      ALP(J,I) = -ALOG(RAT(J))
     1         /AMAX1(VMD(J,I)-VEE(1,J,I),VEE(K,J,I)-VMD(J,I))
   40 CONTINUE
  560 CONTINUE
  793 IF(IN5.GT.0)CALL READE(NEMAX,GIM)
      IF(IN6.GT.0)CALL READE(NEMAX,AIM)
      IF(IN7.LE.0)GO TO 60
```

```
      CALL READE(NEMAX,ALM)
      DO 50 I=1,NEMAX
   50 ALM(I) = 0.5*ALOG(ALM(I))
   60 IF(IN8.LE.0)GO TO 80
      CALL READE(NEMAX,ALH)
      DO 70 I=1,NEMAX
   70 ALH(I) = 0.5*ALOG(ALH(I))
   80 IF(IN9.LE.0)GO TO 90
      CALL READE(8,ATA)
      ATA = ALOG(ATA)
      ATB = -ALOG(ATB)/3.141596
   90 RETURN
      END
$IBFTC IREAD   M94/2,XR7
CREADI *INTEGER INPUT, 24I3 FORMAT
      SUBROUTINE READI(MAX,IDM)
      COMMON/TAPEID/M1     ,M2
      DIMENSION IDM(10000),H(2)
      DO 30 I=1,MAX,24
      MOD = I + 23
      IF(MOD.GT.MAX)GO TO 10
      READ(M1,1000)(IDM(J),J=I,MOD),H
      GO TO 20
   10 JMX = MOD - MAX
      MOD = MAX
      READ(M1,1000)(IDM(J),J=I,MOD),(K,J=1,JMX),H
   20 IF(LABEL(1).GE.0)WRITE(M2,2000)H,(IDM(J),J=I,MOD)
   30 CONTINUE
 1000 FORMAT(24I3,2A4)
 2000 FORMAT(1X,2A4,3H...,24I4)
      RETURN
      END
$IBFTC EREAD   M94/2,XR7
CREADE *FLOATING INPUT, 8E9.0 FORMAT
      SUBROUTINE READE(MAX,ADM)
      COMMON/TAPEID/M1     ,M2
      DIMENSION ADM(10000),H(2)
      DO 30 I=1,MAX,8
      MOD = I + 7
      IF(MOD.GT.MAX)GO TO 10
      READ(M1,1000)(ADM(J),J=I,MOD),H
      GO TO 20
   10 JMX = MOD - MAX
      MOD = MAX
      READ(M1,1000)(ADM(J),J=I,MOD),(X,J=1,JMX),H
   20 IF(LABEL(1).GE.0)WRITE(M2,2000)H,(ADM(J),J=I,MOD)
   30 CONTINUE
```



```
 1000 FORMAT(8E9.0,2A4)
 2000 FORMAT(1X,2A4,3H...,1P8E12.4)
      RETURN
      END
$IBFTC AREAD   M94/2,XR7
CREADA *HOLLERITH INPUT, 18A4 FORMAT
      SUBROUTINE READA(MAX,ADM)
      COMMON/TAPEID/M1     ,M2
      DIMENSION ADM(10000),H(2)
      DO 30 I=1,MAX,18
      MOD = I + 17
      IF(MOD.GT.MAX)GO TO 10
      READ(M1,1000)(ADM(J),J=I,MOD),H
      GO TO 20
   10 JMX = MOD - MAX
      MOD = MAX
      READ(M1,1000)(ADM(J),J=I,MOD),(X,J=1,JMX),H
   20 IF(LABEL(1).GE.0)WRITE(M2,2000)H,(ADM(J),J=I,MOD)
   30 CONTINUE
 1000 FORMAT(20A4)
 2000 FORMAT(1X,2A4,27(1H.),18A4)
      RETURN
      END
$IBFTC ISREAD  M94/2,XR7
      SUBROUTINE READIS(MAX,IDM)
      DIMENSION IDM(10000),JDM(2,12)
      DO 10 I=1,MAX,12
      MOD = MIN0(12,MAX+1-I)
      CALL READI(2*MOD,JDM)
      DO 10 J=1,MOD
      K = JDM(1,J)
   10 IDM(K) = JDM(2,J)
      RETURN
      END
$IBFTC IEREAD  M94/2,XR7
      SUBROUTINE READIE(MAX,IDM,ADM)
      COMMON/TAPEID/M1     ,M2
      DIMENSION IDM(1000),ADM(1000),H(2)
      DO 30 I=1,MAX,6
      MOD = I + 5
      IF(MOD.GT.MAX)GO TO 10
      READ (M1,1000)(IDM(J),ADM(J),J=I,MOD),H
      GO TO 20
   10 JMX = MOD - MAX
      MOD = MAX
      READ(M1,1000)(IDM(J),ADM(J),J=I,MOD),(K,X,J=1,JMX),H
   20 IF(LABEL(1).GE.0)WRITE(M2,2000)H,(IDM(J),ADM(J),J=I,MOD)
```

```
   30 CONTINUE                                                          FAST1221
 1000 FORMAT(6(I3,E9.0),2A4)                                            FAST1222
 2000 FORMAT(1X,2A4,3H...,6(I4,1PE12.4))                                FAST1223
      RETURN                                                            FAST1224
      END                                                               FAST1225
$IBFTC ESREAD  M94/2,XR7                                                FAST1226
      SUBROUTINE READES(MAX,ADM)                                        FAST1227
      DIMENSION ADM(10000),IDM(6),BDM(6)                                FAST1228
      DO 10 I=1,MAX,6                                                   FAST1229
      MOD = MIN0(6,MAX+1-I)                                             FAST1230
      CALL READIE(MOD,IDM,BDM)                                          FAST1231
      DO 10 J=1,MOD                                                     FAST1232
      K = IDM(J)                                                        FAST1233
   10 ADM(K) = BDM(J)                                                   FAST1234
      RETURN                                                            FAST1235
      END                                                               FAST1236
$IBFTC EEREAD  M94/2,XR7                                                FAST1237
CREADEE*ALTERNATING FLOATING INPUT, 8E9.0 FORMAT                        FAST1239
      SUBROUTINE READEE(MAX,ADM,BDM)                                    FAST1240
      DIMENSION ADM(10000),BDM(10000),CDM(2,4)                          FAST1241
      K= 0                                                              FAST1242
      DO 10 I=1,MAX,4                                                   FAST1243
      MOD = MIN0(4,MAX-K)                                               FAST1244
      CALL READE(2*MOD,CDM)                                             FAST1245
      DO 10 J=1,MOD                                                     FAST1246
      K = K + 1                                                         FAST1247
      ADM(K) = CDM(1,J)                                                 FAST1248
   10 BDM(K) = CDM(2,J)                                                 FAST1249
      RETURN.                                                           FAST1250
      END                                                               FAST1251
$IBFTC SREAD   M94/2,XR7                                                FAST1252
CREADS * SURFACE FORMAT, 3I3,7E9.0                                      FAST1253
      SUBROUTINE READS(IDM,ADM)                                         FAST1254
      COMMON/TAPEIO/M1    ,M2                                           FAST1255
      DIMENSION IDM(3),ADM(7),H(2)                                      FAST1256
      READ(M1,1000)IDM,ADM,H                                            FAST1257
      IF(LABEL(1).GE.0)WRITE(M2,2000)H,IDM,ADM                          FAST1258
 1000 FORMAT(3I3,7E9.0,2A4)                                             FAST1259
 2000 FORMAT(1X,2A4,3H...,3I4,1P7E12.4)                                 FAST1260
      RETURN                                                            FAST1261
      END                                                               FAST1262
$IBFTC RREAD   M94/2,XR7                                                FAST1263
CREADR * VOLUME SOURCE FORMAT, 4E9.0,12I3 FORMAT                        FAST1264
      SUBROUTINE READR(IDM,ADM)                                         FAST1265
      COMMON/TAPEIO/M1    ,M2                                           FAST1266
      DIMENSION ADM(4),IDM(12),H(2)                                     FAST1267
      READ(M1,1000)IDM,ADM,H                                            FAST1268
```



```
      IF(LABEL(1).GE.0)WRITE(M2,2000)H,IDM,ADM                          FAST1269
 1000 FORMAT(12I3,4E9.0,2A4)                                            FAST1270
 2000 FORMAT(1X,2A4,3H...,12I4,1P4E12.4)                                FAST1271
      RETURN                                                            FAST1272
      END                                                               FAST1273
$IBFTC FREAD   M94/2,XR7                                                FAST1274
CREADF * RESPONSE FUNCTION FORMAT, 2I3,3A4,6E9.0                        FAST1275
      SUBROUTINE READF(IDM,ADM,BDM)                                     FAST1276
      COMMON/TAPEIO/M1    ,M2                                           FAST1277
      DIMENSION IDM(2),ADM(3),BDM(6),H(2)                               FAST1278
      READ(M1,1000)IDM,ADM,BDM,H                                        FAST1279
      IF(LABEL(1).GE.0)WRITE(M2,2000)H,IDM,ADM,BDM                      FAST1280
 1000 FORMAT(2I3,3A4,6E9.0,2A4)                                         FAST1281
 2000 FORMAT(1X,2A4,3H...,2I4,2X,3A4,2X,1P6E12.4)                       FAST1282
      RETURN                                                            FAST1283
      END                                                               FAST1284
$ORIGIN        ALPHA,REW                                                FAST1285
$IBFTC RSOLVE  M94/2,XR7                                                FAST1286
      SUBROUTINE SOLVIT                                                 FAST1287
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST1238
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST1288
     2              NUNITX,NIMAX ,NMPAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST1289
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1290
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1291
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1292
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST1293
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST1294
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST1295
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST1296
     3              INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST1297
     4              IVOL  ,ICDT  ,IXDT  ,IRSI  ,IAEP  ,IVMD  ,IGIP  ,   FAST1298
     5              IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST1299
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST1300
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST1301
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST1302
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST1303
     1              IFXT  ,IFXE  ,IFXA                                  FAST1304
      COMMON H(1)                                                       FAST1305
      IF(NPOINT.GT.0)GO TO 20                                           FAST1306
      CALL SOBER                                                        FAST1307
     1(NGMAX,1,H(IWS),H(IES),H(IWC),H(IEC),H(IST),H(INRG),H(IMTL),      FAST1308
     2 H(IIDR),H(IIDS),H(IVOL),H(IXDT),H(ICDT),H(IFXP),H(IFXS),H(IXMP), FAST1309
     3H(INSC), H(IALM),H(IALH),H(IRHO),H(IELL),H(ISSH),H(ISGT),NEMOD)   FAST1310
      GO TO 10                                                          FAST1311
   20 CALL SOLVER(1,NEMOD,NGMAX,H(IXCT),H(ICDT),H(IST),H(INRG),H(INSC), FAST1312
     1           H(IMTL),H(IRHO),H(ISSH),H(ISGT),H(IELL),H(IWS),H(IES),FAST1313
     2           H(IWC),H(IEC),H(IXMP),H(IALM),H(IALH),H(IFXP),H(IFXS),FAST1314
```

Astronuclear Laboratory


```
      3H(ISTP),H(INPP),H(INCP),H(IIDR))                                 FAST1315
   10 RETURN                                                            FAST1316
      END                                                               FAST1317
$IBFTC SOBER2  M94/2,XR7                                                FAST1318
CSOBFR2, SOLUTICN OF THE BOLTZMANN ECUATION BY RANDOM SAMPLING          FAST1319
      SUBROUTINE SOBER(L1,L2,WS,ES,WC,EC    ,ST,NRG    ,MTL,IDR,IDS,VOL,FAST1320
     1XDT,CDT,FXP,FXS,XMP,NSC,ALM,ALF,RHO,ELL,SSH,SGT,L3)               FAST1321
      DIMENSION SSH(1),SGT(L3,L2)                                       FAST1322
      COMMON/LIMITS/NSTCRE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST1323
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST1324
     2              NUNITX,NIMAX ,NMMAX ,NCRDER,NDOWN ,INELAS,NTRANS,   FAST1325
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1326
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1327
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1328
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST1329
      COMMON/FLUXES/KKK    ,NTALLY,ERROR ,TCTALN,TOTALE,RHON  ,SNORM ,TOTFAST1330
      COMMON/POINTX/NTOTAL,III    ,MOM                                  FAST1331
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA   ,ATB   ,ATC   ,   FAST1332
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,   FAST1333
     2              KMIN  ,KMAX  ,JBAR  ,NZERO                          FAST1334
      DIMENSION ELL(1)                                                  FAST1335
      DIMENSION WS(1),ES(1),WC(1),EC(1),XMP(1),ST(1),NRG(1),NSC(1),     FAST1336
     1          MTL(1),IDR(1),IDS(1),VOL(1),XDT(3,1),CDT(3,1),          FAST1337
     2          FXP(L1,L2),FXS(L1,L2),GIM(1),AIM(1),ALM(1),ALH(1),RHO(1)FAST1338
      DIMENSION X(3),XC(3),C(3),CC(3),CN(3),CP(3)                       FAST1339
      SNORM = AMAX1(100.0,VECTOR(XCT,BDC,C))                            FAST1340
      NTOTAL = 0                                                        FAST1341
      IOMEGA = 1                                                        FAST1342
      IF(NVDMOD.GT.0) IOMEGA = NUMBER + 1                               FAST1343
      CALL GROUP(BDC)                                                   FAST1344
      DO 250 NNN=1,NPRINT                                               FAST1345
      DO 240 MMM-1,NUNITS                                               FAST1346
      NTALLY = 1                                                        FAST1347
      ERROR = ATD                                                       FAST1348
      RUNTOT = 0.0                                                      FAST1349
      DO 210 KKK=1,KALICE                                               FAST1350
      IF(KKK.GT.1)GO TO 10                                              FAST1351
      IF(MODELP.GT.0)GO TO 5                                            FAST1352
      PDT = PSTAR(NN,X)                                                 FAST1353
      GO TO 40                                                          FAST1354
    5 PDT = SPHERE(BDC,STC,C,NST,ST,NRG,NSC)                            FAST1355
      IF(NST.EQ.0) GO TC 220                                            FAST1356
      NN = NRG(NST)                                                     FAST1357
      DO 6 I=1,3                                                        FAST1358
    6 X(I) = BDC(I) + C(I)*STC                                          FAST1359
      GO TO 40                                                          FAST1360
   10 IF((NTALLP+NTALLY).LE.0)GO TO 220                                 FAST1361
```

Astronuclear Laboratory


```
      IF(TOTALN.EC.0.0)GO TO 13                                         FAST1362
      EBAR = TOTALE/TOTALN                                              FAST1363
      DO 12 I=1,NEMAX                                                   FAST1364
      IF(EBAR.GE.ELL(I+1))GO TO 14                                      FAST1365
   12 CONTINUE                                                          FAST1366
   13 I = (JMIN + JMAX)/2                                               FAST1367
   14 JBAR = I                                                          FAST1369
      PDT = USTAR(STC,BDC,X,C,NST,ST,NRG)                               FAST1370
      IF(NST.EQ.0)GO TO 220                                             FAST1371
      RUNTOT = RUNTOT + STC                                             FAST1372
      NN = NRG(NST)                                                     FAST1373
      PDT = PDT/STC**2                                                  FAST1374
      CALL KERNEL(1,NST,ST,NRG)                                         FAST1375
      DO 20 I=1,3                                                       FAST1376
      X(I) = X(I) + C(I)*STC                                            FAST1377
   20 CC(I) = C(I)                                                      FAST1378
      KMIN = JMIN                                                       FAST1379
      KMAX = JMAX                                                       FAST1380
      DO 30 I=JMIN,JMAX                                                 FAST1381
      WC(I) = PDT*WS(I)*EXP(-XMP(I))                                    FAST1382
   30 EC(I) = ES(I)                                                     FAST1383
   40 NTALLP = NTALLY                                                   FAST1384
      NTALLY = 0                                                        FAST1385
      TOTALN = 0.0                                                      FAST1386
      TOTALF = 0.0                                                      FAST1387
      IF(NPDMAX.EC.0) GC TO 90                                          FAST1368
      NPD = 0                                                           FAST1388
      DO 80 I=1,NDMAX                                                   FAST1389
      IF(IDR(I).GT.0)GO TO 80                                           FAST1390
      NPD = NPD + 1                                                     FAST1391
      STM = VECTOR(X,XDT(1,  I),C)                                      FAST1392
      IF(KKK.GT.1)GO TO 50                                              FAST1393
      CALL SZERC(NN,PCT,X,C)                                            FAST1394
      GO TO 60                                                          FAST1395
   50 CALL SINGLE(NN,COSINE(C,CC),1.0)                                  FAST1396
   60 IF(JMIN.GT.JMAX)GC TO 70                                          FAST1397
      TOT = RUNTOT + STM                                                FAST1398
      CALL PATH(NN,STM,X,C,NST,ST,NRG,NSC)                              FAST1399
      CALL KERNEL(1,NST,ST,NRG)                                         FAST1400
      CALL DETECT(I,-1,1.0/STM**2,COSINE(C,CDT(1,I)),0.0)               FAST1401
   70 IF(NPD.EQ.NPDMAX)GO TO 90                                         FAST1402
   80 CONTINUE                                                          FAST1403
   90 IF(KKK.LT.KALIDE) GO TC 95                                        FAST1404
      IF(NVDMOD.EC.0) GC TC 210                                         FAST1405
   95 IF(KKK.GT.1) GO TC 102                                            FAST1406
      PDX = PDT                                                         FAST1407
      IF(MODELQ.GT.0) GO TO 100                                         FAST1408
```

```
      MGT = 0
      GO TO 108
  100 FST = VECTOR(XCT,X,CC)
      JBAR = JZERO
      GO TO 104
  102 PDX = 1.0
      MGT = 1
  104 FST = RHO(NN)*SSH(JBAR)
      GST = 0.0
      M = MTL(NN)
      IF(M.GT.0) GST = SGT(JBAR,M)
      ALPHA = 0.0
      HST = FST + GST
      IF(HST.GT.0.0) ALPHA = (FST*ALH(JBAR) + GST*ALM(JBAR))/HST
      STM = VECTOR(X,BCC,CP)
      BETA = 0.0
      IF(FST.LE.DELTA) GO TO 108
      CALL PATH(NN,FST-DELTA,X,CP,NST,ST,NRG,NSC)
      XMPT = 0.0
      DO 106 I=1,NST
      J = NRG(I)
      K = MTL(J)
      SIG = RHO(J)*SSH(JBAR)
      IF(K.GT.0) SIG = SIG + SGT(JBAR,K)
  106 XMPT = XMPT + SIG*ST(I)
      IF(XMPT.GT.0.0) BETA = AS*(ALOG(12.0*XMPT))/4.0
      IF(MODELV.EQ.2) BETA = AS*XMPT/2.0
  108 DO 205 LLL=1,IOMEGA
      IF(MGT.GT.0) GO TO 110
      PDA = PDX*QSTAR(NN,X,C)
      GO TO 112
  110 PDA = PDX*VSTAR(CP,CC,C,BETA,BS*ALPHA)
  112 IF(LLL.LT.IOMEGA) PDA = PDA/FLOAT(NUMBER)
      IF(KKK.GT.1) GO TO 114
      CALL SZERO(NN,PDA,X,C)
      GO TO 116
  114 CALL SINGLE(NN,COSINE(C,CC),PDA)
  116 IF(JMIN.GT.JMAX) GO TO 202
      CALL PATH(NN,1.0E+30,X,C,NST,ST,NRG,NSC)
      IF(LLL.EQ.IOMEGA) GO TO 205
      NSTMIN = 1
      STT = 0
      TOT = RUNTOT
      DO 200 N=1,NST
      L = NRG(N)
      M = MTL(L)
      IF(NVDMAX.EQ.0)GO TO 150
```



```
      DO 140 I=1,NDMAX
      IF(IDR(I).NE.L)GO TO 140
      IF(IDS(I).NE.0)GO TO 140
      ANG = COSINE(C,CCT(1,I))
      IF(M.LT.0)GO TO 120
      FST = 1.0/VCL(I)
      RHON = RHO(L)
      GO TO 130
  120 FST = ST(N)/VCL(I)
      TOT = TOT + ST(N)/2.0
  130 CALL KERNEL(NSTMIN,N-1,ST,NRG)
      CALL DETECT(I,M,FST,ANG,ST(N))
      IF(M.LT.0) TOT = TOT - ST(N)/2.0
      NSTMIN = N + 1
      GO TO 150
  140 CONTINUE
  150 STT = STT + ST(N)
      TOT = RUNTOT + STT
      K = NSC(N)
      IF(K*NSDMAX.LE.0)GO TO 200
      MAX = 2 - N/NST
      NGT = 0
      DO 190 M=1,MAX
      DO 160 I=1,NDMAX
      IF(IDR(I).EQ.L.AND.IDS(I).EQ.K)GO TO 170
  160 CONTINUE
      GO TO 190
  170 IF(NGT.GT.0)GO TO 180
      NGT = 1
      CALL NORMAL(K,STT,X,C,CN)
      ANG = COSINE(C,CN)
      CALL KERNEL(NSTMIN,N,ST,NRG)
      NSTMIN = N + 1
  180 FST = 1.0/(ABS(ANG)*VOL(I))
      CALL DETECT(I,-1,FST,ANG,0.0)
  190 L = NRG(N+1)
  200 CONTINUE
  202 IF(LLL.EQ.IOMEGA) GO TO 220
  205 CONTINUE
  210 CONTINUE
  220 CONTINUE
      DO 230 I=1,NDMAX
      DO 230 J=1,NGMAX
      FXS(J,I) = FXS(J,I) + FXP(J,I)**2
  230 FXP(J,I) = 0.0
  235 CONTINUE
  240 CONTINUE
```

```
      CALL ANSWER(NNN)
  250 CONTINUE
      RETURN
      END
$IBFTC SOLVE    M94/2,XR7
CSOLVER*POINT SOLUTION OF THE BOLTZMANN EQUATION*T.M.JORDAN*WANL*1966*
      SUBROUTINE SOLVER(L1,L2,L3,XDT,CDT,ST,NRG,NSC,MTL,RHO,SSH,SGT,ELL,
     1WS,ES,WC,EC,XMP,ALM,ALH,FXP,FXS,STP,NRP,NCP,IDR)
      DIMENSION XT(3),CT(3),C(3),X(3),CC(3),CD(3),XC(3),XP(3)
      DIMENSION XDT(3,1),CDT(3,1),ST(1),NRG(1),MTL(1),RHO(1),SSH(1),
     1          SGT(L2,L1),ELL(1),WS(1),ES(1),WC(1),EC(1),XMP(1),ALM(1),
     2          ALH(1),FXP(L3,L1),FXS(L3,L1),IDR(1)
      COMMON/TAPEID/M1     ,M2
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSCMAX,
     5              NVDMAX,NVDMOD,NPCINT,MODELP,MODELQ,MODELU,MODELV,
     6              NPRINT,NUNITS,NUMBER,KALIDE
      COMMON/FLUXES/KKK    ,NTALLY,ERROR ,TOTALN,TOTALE,RHON  ,SNORM ,TOT
      COMMON/POINTX/NTOTAL,III    ,MOM
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BCC(3),ATA   ,ATB   ,ATC   ,
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,
     2              KMIN  ,KMAX  ,JBAR  ,NZERO
      DIMENSION BDCDUM(3)
      COMMON/ENDRUN/NPDRUN
      NPDRUN = 0
      DO 100 I=1,3
  100 BDCDUM(I) = BDC(I)
      DO 27 III=1,NDMAX
      NTOTAL = 0
      IF(IDR(III).GT.0) GO TO 27
      NPDRUN = NPDRUN + 1
      DO 1 I=1,3
      XT(I) = XDT(I,III)
    1 CT(I) = CDT(I,III)
      CALL GROUP(XT)
      SNORM =AMAX1(100.0,VECTOR(XCT,XT,C))
      DO 30  NNN=1,NPRINT
      DO 29  MMM=1,NUNITS
      IF(MODELP.GT.0) GO TO 93
      NHALF = 1
      PDS = PSTAR(NN,X)
      IF(NN.EQ.0) GO TO 26
      GO TO 94
   93 CONTINUE
```



```
      PDS = SPHERE(XT,STD,C,NST,ST,NRG,NSC)
      IF(NST.EQ.0) GO TO 26
      NN = NRG(NST)
      DO  2 I=1,3
      X(I) = XT(I) + STD*C(I)
    2 C(I) = -C(I)
      NHALF = 2
   94 CONTINUE
      FST = VECTOR(XCT,X,CC)
      NTALLY = 1
      RUNTOT = 0.0
      DO 25  KKK=1,KALIDE
      ERROR = ATD
      NTALLP = NTALLY
      NTALLY = 0
      TOTALN = 0.0
      TOTALE = 0.0
      STD = VECTOR(X,XT,C)
      IF(KKK.GT.1)GO TO 3
      CALL SZERO(NN,PDS,X,C)
      GO TO 4
    3 CALL SINGLE(NN,COSINE(C,CC),1.0)
    4 IF(NHALF.EQ.1)CALL PATH(NN,STD,X,C,NST,ST,NRG,NSC)
      IF(JMIN.GT.JMAX)GO TO 5
      CALL KERNEL(1,NST,ST,NRG)
      TOT = RUNTOT + STD
      CALL DETECT(1   ,-1,1.0/STD**2,COSINE(C,CT),0.0)
    5 IF(KKK.EQ.KALIDE)GO TO 25
      IF((NTALLY + NTALLP).EQ.0)GO TO 26
      IF(TOTALN.EQ.0.0)GO TO 7
      FST = TOTALE/TOTALN
      DO 6 I=1,NEMAX
      IF(FST.GE.ELL(I+1)) GO TO 8
    6 CONTINUE
    7 I = JMIN
    8 JBAR = I
      NGT = 0
      XMPT = 0.0
      STH = STD/2.0
      STT = 0.0
      SIGH = 0.0
      DO 9 I=1,NST
      J = NRG(I)
      SIG = RHO(J)*SSH(JBAR)
      SIGH = SIGH + SIG*ST(I)
      K = MTL(J)
      IF(K.GT.0)SIG = SIG + SGT(JBAR,K)
```

```
      XMPT = XMPT + SIG*ST(I)
      IF(NGT.EQ.1)GO TO 9
      STT = STT + ST(I)
      IF(STT.LT.STH)GO TO 9
      XMPH = XMPT - SIG*(STT - STH)
      NGT = 1
    9 CONTINUE
      BETA = 0.0
      IF(NHALF.EQ.2)XMPH = XMPT - XMPH
      IF(XMPT.GT.0.0)  BETA=AS*(ALOG(12.0*XMPT))/4.0
      IF(MODELV.EQ.2) BETA = AS*XMPT/2.0
      NNC = NN
      IF((KKK+MODELQ).EQ.1) GO TO 10
      IF(XMPT.EQ.0.0)GO TO 1C
      IF(XMPH.LT.XMPT)GO TO 11
   10 PDF = 1.0
      NGT = 0
      GO TO 12
   11 IF(XMPH.EQ.0.0)GO TC 14
      PHALF = XMPH/XMPT
      IF(RANNO(NMB).GT.PHALF)GO TO 15
      PDF = 1.0/PHALF
      NGT = 1
   12 NHALF = 1
      DO 13 I=1,3
      CD(I) = C(I)
      BOC(I) = XT(I)
   13 XC(I) = X(I)
      GO TO 18
   14 PDF = 1.0
      NGT = 0
      GO TO 16
   15 PDF = 1.0/(1.0 -PHALF)
      NGT = 1
   16 NHALF = 2
      DO 17 I=1,3
      XC(I) = X(I)
      CD(I) = -C(I)
      BOC(I) = X(I)
   17 X(I) = XT(I)
      NN = NZERO
   18 ALPHA = 0.0
      IF(XMPT.GT.0.0)ALPHA=(SIGH*ALH(JBAR)+(XMPT-SIGH)*ALN(JBAR))/XMPT
      IF((KKK+MODELQ).GT.1) GO TO 300
      PDA = QSTAR(NN,X,C)
      GO TO 301
  300 PDA = VSTAR(CD,CC,C,BETA,BS*ALPHA)
```



```
  301 CONTINUE
      STM = 1.0E+30
      IF(NGT.EQ.0)GO TO 19
      PSI = COSINE(C,CD)
      IF(PSI.GT.0.0)STM = STH/PSI
   19 CALL PATH(NN,STM,X,C,NST,ST,NRG,NSC)
      PDT = USTAR(STC,BOC,X,C,NST,ST,NRG)
      IF(NST.EQ.0)GO TO 26
      NNP = NRG(NST)
      DO 20 I=1,3
      XP(I) = X(I) + STC*C(I)
   20 X(I) = XC(I)
      STM = VECTOR(X,XP,C)
      NN = NNC
      PDX = PDF*PDA*PDT/STM**2
      IF(KKK.GT.1)GO TO 21
      CALL SZERO(NN,PDS,X,C)
      GO TO 22
   21 CALL SINGLE(NN,COSINE(C,CC),1.C)
   22 IF(JMIN.GT.JMAX)GO TO 26
      IF(NHALF.EQ.2)GO TO 200
      CALL KERNEL(1,NST,ST,NRG)
      GO TO 201
  200 CALL PATH(NN,STM,X,C,NSTP,STP,NRP,NCP)
      CALL KERNEL(1,NSTP,STP,NRP)
  201 CONTINUE
      DO 23 I=JMIN,JMAX
      EC(I) = ES(I)
   23 WC(I) = PDX*WS(I)*EXP(-XMP(I))
      KMIN = JMIN
      KMAX = JMAX
      NN = NNP
      RUNTOT = RUNTOT + STM
      DO 24 I=1,3
      X(I) = XP(I)
   24 CC(I) = C(I)
   25 CONTINUE
   26 CONTINUE
      DO 28 I=1,NDMOD
      DO 28 J=1,NGMAX
      FXS(J,I) = FXS(J,I) + FXP(J,I)**2
   28 FXP(J,I) = 0.0
  277 CONTINUE
   29 CONTINUE
      CALL ANSWER(NNN)
   30 CONTINUE
   27 CONTINUE
```


Astronuclear Laboratory

```
      DO 101 I=1,3                                                      FAST1686
  101 BOC(I) = BOCDUM(I)                                                FAST1687
      RETURN                                                            FAST1688
      END                                                               FAST1689
$IBFTC GROUPI  M94/2,XR7                                                FAST1690
      SUBROUTINE GROUP(X)                                               FAST1691
      COMMON H(1)                                                       FAST1692
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST1693
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST1694
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST1695
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1696
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1697
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1698
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST1699
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST1700
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST1701
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST1702
     3              INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST1703
     4              IVOL  ,ICDT  ,IXCT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST1704
     5              IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST1705
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST1706
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST1707
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST1708
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST1709
     1              IFXT  ,IFXE  ,IFXA                                  FAST1710
      CALL GRODUM(X,1,NEMAX,H(IST),H(INRG),H(INSC),H(IISV),             FAST1711
     1H(IWS),H(IES),H(IJSN),H(IJSX),H(ISPW),H(ISPE),H(IXMP),H(IAIM),    FAST1712
     2H(IGIM),H(IELL))                                                  FAST1713
      RETURN                                                            FAST1714
      END                                                               FAST1715
$IBFTC GRODMI  M94/2,XR7                                                FAST1716
      SUBROUTINE GRODUM(X,L1,L2,ST,NRG,NSC,ISV    ,WS,ES,JSN,JSX,SPW,   FAST1717
     1SPE,XMP,AIM,GIM,ELL)                                              FAST1718
      COMMON/INDEXG/JZERO                                               FAST1719
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BOC(3),ATA   ,ATB   ,ATC   ,   FAST1720
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,   FAST1721
     2              KMIN  ,KMAX  ,JBAR  ,NZERO                          FAST1722
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST1723
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST1724
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST1725
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1726
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1727
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1728
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST1729
      DIMENSION NRG(1),ISV(1)       ,WS(1),ES(1),JSN(1),JSX(1),         FAST1730
     1SPW(L2,L1),SPE(L2,L1),XMP(1),AIM(1),GIM(1),ELL(1)                 FAST1731
      DIMENSION C(3)                                                    FAST1732
```



```
      NZERO = LOCATE(NGS,X)                                             FAST1733
      STM = VECTOR(X,XCT,C)                                             FAST1734
      CALL PATH(NZERO,STM,X,C,NST,ST,NRG,NSC)                           FAST1735
      I = NST + 1                                                       FAST1736
      DO  1 K=1,NST                                                     FAST1737
      I = I - 1                                                         FAST1738
      L = NRG(I)                                                        FAST1739
      J = ISV(L)                                                        FAST1740
      IF(J.GT.0) GO TO 2                                                FAST1741
    1 CONTINUE                                                          FAST1742
      JZERO = NEMOD/2                                                   FAST1743
      GO TO 100                                                         FAST1744
    2 NTT = I                                                           FAST1745
C CALCULATE AVERAGE SOURCE GROUP INDEX                                  FAST1746
      TOTN = 0.0                                                        FAST1747
      TOTE = 0.0                                                        FAST1748
      MIN = JSN(J)                                                      FAST1749
      MAX = JSX(J)                                                      FAST1750
      JMIN = MIN                                                        FAST1751
      JMAX = MAX                                                        FAST1752
      DO 10 I=MIN,MAX                                                   FAST1753
      WS(I) = SPW(I,J)                                                  FAST1754
   10 ES(I) = SPE(I,J)                                                  FAST1755
      IF(NTT.EQ.1)GO TO 30                                              FAST1756
      CALL KERNEL(1,NTT-1,ST,NRG)                                       FAST1757
      IF(XMP(MIN).EQ.0.0)GO TO 30                                       FAST1758
      DO 20 I=MIN,MAX                                                   FAST1759
   20 WS(I) = WS(I)*EXP(-XMP(I))*(1.0 + AIM(I)*XMP(I))                  FAST1760
   30 CALL KERNEL(1,0,ST,NRG)                                           FAST1761
      CALL KERNEL(NTT,NTT,ST,NRG)                                       FAST1762
      DO 40 I=MIN,MAX                                                   FAST1763
      FST = GIM(I)*WS(I)                                                FAST1764
      IF(XMP(I).NE.0.0)FST = FST/XMP(I)                                 FAST1765
      TOTN = TOTN + FST                                                 FAST1766
   40 TOTE = TOTE + FST*ES(I)                                           FAST1767
      EBAR = TOTE/TOTN                                                  FAST1768
      DO 50 I=MIN,MAX                                                   FAST1769
      IF(EBAR.GE.ELL(I+1))GO TO 60                                      FAST1770
   50 CONTINUE                                                          FAST1771
      I = (MIN + MAX)/2                                                 FAST1772
   60 JZERO = I                                                         FAST1773
  100 RETURN                                                            FAST1774
      END                                                               FAST1775
$IBFTC SZEROF  M94/2,XR7                                                FAST1776
CSZERO *FIXED SOURCE EVALUATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966*FAST1777
      SUBROUTINE SZERO(NNN,PDF,XXX,CCC)                                 FAST1778
      DIMENSION XXX(1),CCC(1)                                           FAST1779
```

```
      COMMON         H(1)                                              FAST1780
      COMMON/DUMSZE/NN    ,PCS   ,X(3)  ,C(3)                          FAST1781
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,  FAST1782
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,  FAST1783
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,  FAST1784
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,  FAST1785
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPOMAX,NPOMOD,NSOMAX,  FAST1786
     5              NVOMAX,NVOMCD,NPGINT,MODELP,MODELQ,MODELU,MODELV,  FAST1787
     6              NPRINT,NUNITS,NUMBER,KALIDE                        FAST1788
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,INTL  ,IRHO  ,IXR   ,IELL  ,  FAST1789
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,  FAST1790
     2              IXTR  ,ISUV  ,IAIN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,  FAST1791
     3              INTG  ,IELF  ,IFGW  ,IIDS  ,IIOV  ,IIOR  ,IIOS  ,  FAST1792
     4              IVOL  ,ICCT  ,IXCT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,  FAST1793
     5              IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,  FAST1794
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,  FAST1795
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,  FAST1796
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,  FAST1797
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,  FAST1798
     1              IFXT  ,IFXE  ,IFXA                                 FAST1799
      COMMON/REGSCA/NSRMAX,INSR                                        FAST1800
      INSR = 0                                                         FAST1801
      NN = NNN                                                         FAST1802
      PDS = PDF                                                        FAST1803
      DO 10 I=1,3                                                      FAST1804
      X(I) = XXX(I)                                                    FAST1805
   10 C(I) = CCC(I)                                                    FAST1806
      CALL SZEDUM(NXMAX,5,1,NEMAX,H(IISV),H(IXTR),H(INSG),H(IJSN),     FAST1807
     1           H(IJSX),H(INPC),H(IVEE),H(IVAL),H(ISPW),H(ISPE),      FAST1808
     2H(IWS),H(IES),H(IIDV))                                           FAST1809
      RETURN                                                           FAST1810
      END                                                              FAST1811
$IBFTC SZEDM1  M94/2,XR7                                               FAST1812
CSZEDUM*FIXED SOURCE EVALUATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966*FAST1813
      SUBROUTINE SZEDUM(L1,L2,L3,L4,ISV,XTR,NSG,JSN,JSX,NPC,VEE,VAL,SPW,FAST1814
     1SPE,WS,ES,IDV)                                                   FAST1815
      DIMENSION IDV(1)                                                 FAST1816
      DIMENSION Z(3),D(3),V(5),ROT(3,3)                                FAST1817
      DIMENSION ISV(1),XTR(3,1),NSG(1),JSN(1),JSX(1),NPC(5,1),ES(1),   FAST1818
     1         VEE(L1,L2,L3),VAL(L1,L2,L3),SPW(L4,L3),SPE(L4,L3),WS(1) FAST1819
      EQUIVALENCE (Z(1),Z1),(Z(2),Z2),(Z(3),Z3),(D(1),D1),(D(2),D2),   FAST1820
     1 (D(3),D3),(V(1),V1),(V(2),V2),(V(3),V3),(V(4),V4),(V(5),V5)     FAST1821
      COMMON/DUMSZE/NN    ,PCS   ,X(3)  ,C(3)                          FAST1822
      COMMON/POINTX/NTOTAL,III   ,MOM                                  FAST1823
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,  FAST1824
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,  FAST1825
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,  FAST1826
```



```
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,  FAST1827
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPOMAX,NPOMOD,NSOMAX,  FAST1828
     5              NVOMAX,NVOMCD,NPCINT,MODELP,MODELQ,MODELU,MODELV,  FAST1829
     6              NPRINT,NUNITS,NUMBER,KALIDE                        FAST1830
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA   ,ATB   ,ATC   ,  FAST1831
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,  FAST1832
     2              KMIN  ,KMAX  ,JBAR  ,NZERO                         FAST1833
      COMMON/LLSREG/LLREG                                              FAST1834
      COMMON/SPASHL/PSORS                                              FAST1835
      IF(NN.GT.0) N = ISV(NN)                                          FAST1837
      N = LLREG                                                        FAST1838
      MOM = 0                                                          FAST1839
      IF(NVMOD.EQ.0) GO TO 3                                           FAST1840
      DO 1 I=1,NVMOD                                                   FAST1841
      IF(IDV(I).EQ.N) GO TO 2                                          FAST1842
    1 CONTINUE                                                         FAST1843
      GO TO 3                                                          FAST1844
    2 MOM = I                                                          FAST1845
    3 CONTINUE                                                         FAST1846
C UN-TRANSLATE                                                         FAST1847
      DO 10 I=1,3                                                      FAST1848
   10 Z(I) = X(I) - XTR(I,N)                                           FAST1849
C SOURCE GEOMETRY CHECK                                                FAST1850
      IF(NSG(N).GT.0)GO TO 30                                          FAST1851
C RECTANGULAR                                                          FAST1852
      DO 20 I=1,3                                                      FAST1853
C SPATIAL VARIABLES                                                    FAST1854
      V(I) = Z(I)                                                      FAST1855
C DIRECTION COSINES                                                    FAST1856
   20 D(I) = C(I)                                                      FAST1857
      IF(NN) 105,105,100
C CYLINDRICAL AND SPHERICAL, AZIMUTHAL ANGLE                           FAST1859
   30 V2 = 0.0                                                         FAST1860
      IF((ABS(Z1) + ABS(Z2)).GT.0.0) V2 = ATAN2(Z2,Z1)                 FAST1861
C RADIUS NEXT                                                          FAST1862
      MAX = NSG(N) + 1                                                 FAST1863
      V1 = 0.0                                                         FAST1864
      DO 40 I=1,MAX                                                    FAST1865
   40 V1 = V1 + Z(I)**2                                                FAST1866
      V1 = SQRT(V1)                                                    FAST1867
      IF(MAX.EQ.3)GO TO 50                                             FAST1868
C CYLINDRICAL, Z-COORDINATE                                            FAST1869
      V3 = Z3                                                          FAST1870
      IF(NN.LE.0) GO TO 105
C ROTATION SETUP, POLAR                                                FAST1871
      R = V1                                                           FAST1872
      CPH = 1.0                                                        FAST1873
```

```
      SPH = 0.0                                                         FAST1874
      GO TO 60                                                          FAST1875
C SPHERICAL, COSINE OF POLAR ANGLE                                      FAST1876
   50 V3 = Z3/V1                                                        FAST1877
      IF(NN.LE.0) GO TO 105
C ROTATION SETUP, POLAR                                                 FAST1878
      CPH = V3                                                          FAST1879
      SPH = SQRT(1.0 - CPH**2)                                          FAST1880
      R = V1*SPH                                                        FAST1881
C ROTATION SETUP, AZIMUTHAL                                             FAST1882
   60 IF(R.GT.0.0)GO TO 70                                              FAST1883
      CTH = 1.0                                                         FAST1884
      STH = 0.0                                                         FAST1885
      GO TO 80                                                          FAST1886
   70 CTH = Z1/R                                                        FAST1887
      STH = Z2/R                                                        FAST1888
C CALCULATE ROTATION MATRIX                                             FAST1889
   80 CALL ROTATE(CPH,SPH,CTH,STH,ROT)                                  FAST1890
C ROTATED DIRECTION COSINES                                             FAST1891
      DO 90 I=1,3                                                       FAST1892
      D(I) = 0.0                                                        FAST1893
      DO 90 J=1,3                                                       FAST1894
   90 D(I) = D(I) + ROT(I,J)*C(J)                                       FAST1895
C COSINE OF POLAR ANGLE, ANGULAR PDF                                    FAST1896
  100 V5 = D3                                                           FAST1897
C AZIMUTHAL ANGLE, ANGULAR PDF                                          FAST1898
      V4 = 0.0                                                          FAST1899
      IF((ABS(D1) + ABS(D2)).GT.0.0) V4 = ATAN2(D2,D1)                  FAST1900
C ALL SOURCE VARIABLES DEFINED, SET ERROR INDICATOR, TOTAL PDF          FAST1901
      JMIN = JSN(N)                                                     FAST1902
      JMAX = 0                                                          FAST1903
  105 CONTINUE
      DO 130 I=1,5                                                      FAST1904
C CHECK FOR DELTA FUNCTION, ASSUMED CORRECT                             FAST1905
      IF(NPC(I,N).EQ.1)GO TO 130                                        FAST1906
C NOT DELTA FUNCTION, CHECK RANGE                                       FAST1907
      K = NPC(I,N)                                                      FAST1908
      IF(((V(I) - VEE(1,I,N))*(VEE(K,I,N) - V(I))).LT.0.0)GO TO 150     FAST1909
C INSIDE RANGE, FIND INTERVAL                                           FAST1910
      DO 110 J=2,K                                                      FAST1911
      IF(VEE(J,I,N).GE.V(I))GO TO 120                                   FAST1912
  110 CONTINUE                                                          FAST1913
C LINEAR INTERPOLATION                                                  FAST1914
  120 K = J - 1                                                         FAST1915
      PDS=PDS*(VAL(J,I,N)*(V(I)-VEE(K,I,N))+VAL(K,I,N)*(VEE(J,I,N)-V(I))FAST1916
     1        )/(VEE(J,I,N)-VEE(K,I,N))                                 FAST1917
      IF(NN.GT.0) GO TO 130                                             FAST1918
```



```
      IF(I.LT.3) GO TO 130                                              FAST1919
      PSORS = PDS                                                       FAST1920
      GO TO 150                                                         FAST1921
  130 CONTINUE                                                          FAST1922
C DEFINE NUMBER IN GROUP AND AVERAGE ENERGY                             FAST1923
      JMAX = JSX(N)                                                     FAST1924
      DO 140 I=JMIN,JMAX                                                FAST1925
      WS(I) = PDS*SPW(I,N)                                              FAST1926
  140 ES(I) = SPE(I,N)                                                  FAST1927
C REGULAR AND OUTSIDE-RANGE RETURNS                                     FAST1928
  150 RETURN                                                            FAST1929
      END                                                               FAST1930
$IBFTC KERNLE  M94/2,XR7                                                FAST1931
CKERNEL*MEAN FREE PATH CALCULATION FOR PROGRAM FASTER*T.H.JORDAN*WANL*66FAST1932
      SUBROUTINE KERNEL(MIN,MAX,ST,NRG)                                 FAST1933
      COMMON        H(1)                                                FAST1934
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST1935
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST1936
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NODWN ,INELAS,NTRANS,   FAST1937
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1938
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1939
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1940
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST1941
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST1942
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST1943
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST1944
     3              INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST1945
     4              IVOL  ,ICDT  ,IXCT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST1946
     5              IAIM  ,IALH  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST1947
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST1948
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST1949
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST1950
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST1951
     1              IFXT  ,IFXE  ,IFXA                                  FAST1952
      CALL KERDUM(NEMOD,1,H(IXMP),H(IAE),H(IBE),H(IELL),H(IELW),H(IES), FAST1953
     1            H(IRHO),H(ISSH),H(IMTL),H(ISGT),MIN,MAX,ST,NRG)       FAST1954
      RETURN                                                            FAST1955
      END                                                               FAST1956
$IBFTC KERDM1  M94/2,XR7                                                FAST1957
CKERDUM*MEAN FREE PATH CALCULATION FOR PROGRAM FASTER*T.H.JORDAN*WANL*66FAST1958
      SUBROUTINE KERDUM(L1,L2,XMP,AE,BE,ELL,ELW,ES,RHO,SSH,MTL,SGT,MIN, FAST1959
     1                  MAX,ST,NRG)                                     FAST1960
      DIMENSION XMP(1),AE(1),BE(1),ELL(1),ELW(1),ES(1),NRG(1),MTL(1),   FAST1961
     1          ST(1),RHO(1),SGT(L1,L2),SSH(1)                          FAST1962
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BOC(3),ATA   ,ATB   ,ATC   ,   FAST1963
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,   FAST1964
     2              KMIN  ,KMAX  ,JBAR  ,NZERO                          FAST1965
```

Astronuclear Laboratory

```
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,    FAST1966
     1               NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST1967
     2               NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST1968
     3               NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST1969
     4               NOMAX ,NGMJD ,MOMENT,NOMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST1970
     5               NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST1971
     6               NPRINT,NUNITS,NUMBER,KALIDE                         FAST1972
      FST = 0.0                                                          FAST1973
      IF(MIN.GT.1)GC TO 30                                               FAST1974
      DO 10 I=JMIN,JMAX                                                  FAST1975
   10 XMP(I) = 0.0                                                       FAST1976
      IF(NXSECT.GT.0)GO TO 30                                            FAST1977
      DO 20 I=JMIN,JMAX                                                  FAST1978
      AE(I) = (ES(I) - ELL(I+1))/ELW(I)                                  FAST1979
   20 BE(I) = (ELL(I) - ES(I))/ELW(I)                                    FAST1980
   30 IF(MIN.GT.MAX)GO TO 90                                             FAST1981
      DO 70 I=MIN,MAX                                                    FAST1982
      J = NRG(I)                                                         FAST1983
      IF(MTL(J)) 70,60,40                                                FAST1984
   40 K = MTL(J)                                                         FAST1985
      DO 50 L=JMIN,JMAX                                                  FAST1986
   50 XMP(L) = XMP(L) + ST(I)*(AE(L)*SGT(L,K) + BE(L)*SGT(L+1,K))        FAST1987
   60 FST = FST + ST(I)*RHO(J)                                           FAST1988
   70 CONTINUE                                                           FAST1989
      IF(FST.EQ.0.0)GO TO 90                                             FAST1990
      DO 80 I=JMIN,JMAX                                                  FAST1991
   80 XMP(I) = XMP(I) + FST*(AE(I)*SSH(I) + BE(I)*SSH(I+1))              FAST1992
   90 RETURN                                                             FAST1993
      END                                                                FAST1994
$IBFTC SINGEL  M94/2,XR7                                                 FAST1995
CSINGLE*SINGLE SCATTERING CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WANLFAST1996
      SUBROUTINE SINGLE(NNN,AAA,BBB)                                     FAST1997
      COMMON        H(1)                                                 FAST1998
      COMMON/DUMSIN/NN     ,PSI    ,PDA                                  FAST1999
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,    FAST2000
     1               NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST2001
     2               NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST2002
     3               NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2003
     4               NDMAX ,NGMOD ,MOMENT,NOMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST2004
     5               NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2005
     6               NPRINT,NUNITS,NUMBER,KALIDE                         FAST2006
      COMMON/INDEXS/INTP   ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST2007
     1               IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST2008
     2               IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST2009
     3               INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST2010
     4               IVOL  ,ICOT  ,IXOT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST2011
     5               IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST2012
```

```
     6               ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIOE  ,IIDI  ,   FAST2013
     7               ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST2014
     8               IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST2015
     9               INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST2016
     1               IFXT  ,IFXE  ,IFXA                                  FAST2017
      COMMON/REGSCA/NSRMAX,INSR                                          FAST2018
      DIMENSION N(1)                                                     FAST2019
      EQUIVALENCE (N(1),H(1))                                            FAST2020
      NN = NNN                                                           FAST2021
      PSI = AAA                                                          FAST2022
      PDA = BBB                                                          FAST2023
      INSR = 0                                                           FAST2024
      IF(NSRMAX.LE.0) GO TO 10                                           FAST2025
      DO 5 I=1,NSRMAX                                                    FAST2026
      IF(NN.EQ.N(I)) INSR = I                                            FAST2027
    5 CONTINUE                                                           FAST2028
   10 CONTINUE                                                           FAST2029
      CALL SINDUM(NDOWN,NORDER,1,INELAS,NEMAX,H(IMTL),H(IRHO),H(IIDE),   FAST2030
     1            H(IIDI)        ,H(ISGS),H(IESB),H(ISSH),H(IDEN),       FAST2031
     2            H(IWS),H(IES),H(IWC),H(IEC),H(IELL))                   FAST2032
      RETURN                                                             FAST2033
      END                                                                FAST2034
$IBFTC SINDMI  M94/2,XR7                                                 FAST2035
CSINDUM*SINGLE SCATTERING CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WANLFAST2036
      SUBROUTINE SINDUM(L1,L2,L3,L4,L5,MTL,RHO,IDE,IDI     ,SGS,ESB,SSH, FAST2037
     1                  DEN,WS,ES,WC,EC,ELL)                             FAST2038
      DIMENSION MTL(1),RHO(1),IDE(L1,L2,L3),IDI(L4,L3),SGS(1 ,1 ),ESB(1)FAST2039
     1          ,SSH(1),DEN(1),WS(1),ES(1),WC(1),EC(1),ELL(1),KMX(1)     FAST2040
      COMMON/DUMSIN/NN     ,PSI    ,PDA                                  FAST2041
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA   ,ATB   ,ATC   ,    FAST2042
     1               ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,   FAST2043
     2               KMIN  ,KMAX  ,JBAR  ,NZERO                          FAST2044
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,    FAST2045
     1               NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST2046
     2               NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST2047
     3               NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2048
     4               NOMAX ,NGMOD ,MOMENT,NOMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST2049
     5               NVDMAX,NVDMCD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2050
     6               NPRINT,NUNITS,NUMBER,KALIDE                         FAST2051
      FST = 0.0                                                          FAST2052
      M = MTL(NN)                                                        FAST2053
      JMIN = NEMAX + 1                                                   FAST2054
      JMAX = 0                                                           FAST2055
      DO 10 I=1,NEMAX                                                    FAST2056
      WS(I) = 0.0                                                        FAST2057
   10 ES(I) = 0.0                                                        FAST2058
      IF(M.LT.0) GO TO 920                                               FAST2059
```

```
      IF(NXSECT.NE.1)GO TO 220                                          FAST2060
  100 PSIN = PDA                                                        FAST2061
      IF(M.EQ.0)GO TO 175                                               FAST2062
      PLM2 = 0.0                                                        FAST2063
      PLM1 = 1.0                                                        FAST2064
      DO 139 L=1,NORDER                                                 FAST2065
      DO 120 K=1,NDOWN                                                  FAST2066
      IF(IDE(K,L,M).LE.0)GO TO 130                                      FAST2067
      JMIN = KMIN                                                       FAST2068
      N = IDE(K,L,M)                                                    FAST2069
      KMOD = MINO(KMAX,NEMAX + 1 - K)                                   FAST2070
      IF(KMIN.GT.KMOD)GO TO 130                                         FAST2071
      I = KMIN + K - 2                                                  FAST2072
      DO 110 J=KMIN,KMOD                                                FAST2073
      I = I + 1                                                         FAST2074
  110 WS(I) = WS(I) + WC(J)*SGS(J,N)*PSIN                               FAST2075
      JMAX = MAXO(JMAX,I)                                               FAST2076
  120 CONTINUE                                                          FAST2077
  130 PLMZ=(PSI*PLM1*FLOAT(2*L-1)-PLM2*FLOAT(L-1))/FLOAT(L)              FAST2078
      PLMZ = PLM1                                                       FAST2079
      PLM1 = PLMZ                                                       FAST2080
      PSIN = PDA*PLMZ                                                   FAST2081
  139 CONTINUE                                                          FAST2082
      KMOD = MINO(KMAX,INELAS)                                          FAST2083
      IF(KMOD.LT.KMIN) GO TO 160                                        FAST2084
      DO 150 J=KMIN,KMOD                                                FAST2085
      K = ID1(J,M)                                                      FAST2086
      IF(K.EQ.0) GO TO 160                                              FAST2087
      N = K/1000                                                        FAST2088
      MAX = MINO(NEMOD-J,K-1000*N)                                      FAST2089
      I = J - 1                                                         FAST2090
      DO 140 K=1,MAX                                                    FAST2091
      I = I + 1                                                         FAST2092
  140 WS(I) = WS(I) + WC(J)*SGS(K,N)*PDA                                FAST2093
      JMAX = MAXO(I,JMAX)                                               FAST2094
  150 CONTINUE                                                          FAST2095
  160 DO 170 I=JMIN,JMAX                                                FAST2096
  170 ES(I) = ES8(I)*WS(I)                                              FAST2097
      FST = DEN(M)                                                      FAST2098
  175 FST = 4.0*PSI*PDA*(FST + RHO(NN))                                 FAST2099
      IF(FST.LE.0.0)GO TO 900                                           FAST2100
      RAT = PSI**2                                                      FAST2101
      MIN = KMIN                                                        FAST2102
      DO 200 J=KMIN,KMAX                                                FAST2103
      E = RAT*EC(J)                                                     FAST2104
      DO 180 I=MIN,NEMAX                                                FAST2105
      IF(E.GE.ELL(I+1))GO TO 190                                        FAST2106
```



```
  180 CONTINUE                                                          FAST2107
      GO TO 210                                                         FAST2108
  190 MIN = I                                                           FAST2109
      JMIN = MINO(JMIN,I)                                               FAST2110
      WSC = FST*SSH(J)*WC(J)                                            FAST2111
      WS(I) = WS(I) + WSC                                               FAST2112
  200 ES(I) = ES(I) + WSC*E                                             FAST2113
  210 JMAX = MAXO(MIN,JMAX)                                             FAST2114
      GO TO 900                                                         FAST2115
C     COMPTON SCATTERING                                                FAST2116
  220 MIN = KMIN                                                        FAST2117
      PSIM = 1.0 - PSI                                                  FAST2118
      PSIS = PSI**2 - 1.0                                               FAST2119
      GST = RHO(NN)*0.49875                                             FAST2120
      IF(M.GT.0)GST = GST + DEN(M)                                      FAST2121
      GST = PDA*GST                                                     FAST2122
      DO 250 J=KMIN,KMAX                                                FAST2123
      RAT = 0.511/(0.511 + PSIM*EC(J))                                  FAST2124
      E = RAT*EC(J)                                                     FAST2125
      DO 230 I=MIN,NEMAX                                                FAST2126
      IF(E.GE.ELL(I+1))GO TO 240                                        FAST2127
  230 CONTINUE                                                          FAST2128
      GO TO 260                                                         FAST2129
  240 MIN = I                                                           FAST2130
      JMIN = MINO(JMIN,I)                                               FAST2131
      FST = WC(J)*GST*RAT**2*(RAT + 1.0/RAT + PSIS)                     FAST2132
      WS(I) = WS(I) + FST                                               FAST2133
      ES(I) = ES(I) + FST*E                                             FAST2134
  250 CONTINUE                                                          FAST2135
  260 JMAX = MIN                                                        FAST2136
  900 IF(JMIN.GT.JMAX)GO TO 920                                         FAST2137
      DO 910 I=JMIN,JMAX                                                FAST2138
      IF(WS(I).NE.0.0) ES(I) = ES(I)/WS(I)                              FAST2139
  910 CONTINUE                                                          FAST2140
  920 RETURN                                                            FAST2141
      END                                                               FAST2142
$IBFTC TRACE    M94/2,XR7                                               FAST2143
CTRACER*PARTIAL PATH LENGTHS FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966*   FAST2144
      SUBROUTINE PATH  (NNN,SSS,XXX,CCC,MMM,ST,NRG,NSC)                 FAST2145
      COMMON           H(1)                                             FAST2146
      COMMON/DUMTRA/NN     ,STM   ,NST   ,X(9)  ,C(9)  ,CX(9)           FAST2147
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST2148
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST2149
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST2150
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2151
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPOMAX,NPCMOD,NSOMAX,   FAST2152
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2153
```

```
      6             NPRINT,NUNITS,NUMBER,KALIDE                                 FAST2154
       COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,          FAST2155
      1             IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,          FAST2156
      2             IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,          FAST2157
      3             INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,          FAST2158
      4             IVOL  ,ICDT  ,IXDT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,          FAST2159
      5             IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,          FAST2160
      6             ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,          FAST2161
      7             ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,          FAST2162
      8             IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,          FAST2163
      9             INRG  ,INSC  ,ISTP  ,IHRP  ,INCP  ,IFXP  ,IFXS  ,          FAST2164
      1             IFXT  ,IFXE  ,IFXA                                         FAST2165
       DIMENSION XXX(1),CCC(1)                                                  FAST2166
       NN = NNN                                                                 FAST2167
       STM = SSS                                                                FAST2168
       DO 10 I=1,3                                                              FAST2169
       X(I) = XXX(I)                                                            FAST2170
    10 C(I) = CCC(I)                                                            FAST2171
       CALL TRADUM(NAMAX,NBMAX,1,H(INTP),H(IAZ),H(IA),H(INS),H(IND),            FAST2172
      1            H(IU),H(IV),H(IW),H(ISI),ST,NRG,NSC)                         FAST2173
       NNN = NST                                                                FAST2174
       RETURN                                                                   FAST2175
       END                                                                      FAST2176
$IBFTC TRADM1  M94/2,XR7                                                        FAST2177
CTRADUM*PARTIAL PATH LENGTHS FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966*           FAST2178
       SUBROUTINE TRADUM(L1,L2,L3,NTP,AZ,A,NS,ND,U,V,W,SI,ST,NRG,NSC)           FAST2179
       DIMENSION NTP(1),AZ(1),A(L1,L3),NS(L2,L3),ST(1),NRG(1),NSC(1),           FAST2180
      1          ND(1),U(1),V(1),W(1),SI(2,1)                                   FAST2181
       COMMON/DUMTRA/NN    ,STM   ,NST   ,X(9)  ,C(9)  ,CX(9)                   FAST2182
       COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,          FAST2183
      1             NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,          FAST2184
      2             NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,          FAST2185
      3             NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,          FAST2186
      4             NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSOMAX,          FAST2187
      5             NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,          FAST2188
      6             NPRINT,NUNITS,NUMBER,KALIDE                                 FAST2189
       DO 110 I=1,3                                                             FAST2190
       J = I + 1 - 3*(I/3)                                                      FAST2191
       X(I+3) = X(I)**2                                                         FAST2192
       X(I+6) = X(I)*X(J)                                                       FAST2193
       C(I+3) = 2.0*C(I)*X(I)                                                   FAST2194
       C(I+6) = C(I)*X(J) + C(J)*X(I)                                           FAST2195
       CX(I+3) = C(I)**2                                                        FAST2196
   110 CX(I+6) = C(I)*C(J)                                                      FAST2197
       DO 120 I=1,NSMAX                                                         FAST2198
   120 ND(I) = 0                                                                FAST2199
       II = NN                                                                  FAST2200
```



```
       STT = 0.0                                                                FAST2201
       DO 410 N=1,NSTMAX                                                        FAST2202
       DO 400 M=1,3                                                             FAST2203
       GO TO(130,140,160),M                                                     FAST2204
   130 IMIN = II                                                                FAST2205
       JJ = 0                                                                   FAST2206
       SP = 1.0E+38                                                             FAST2207
       GO TO 150                                                                FAST2208
   140 IF(NXT.EQ.0)GO TO 400                                                    FAST2209
       IMIN = NXT                                                               FAST2210
   150 IMAX = IMIN                                                              FAST2211
       GO TO 170                                                                FAST2212
   160 IMIN = 1                                                                 FAST2213
       IMAX = NRMAX                                                             FAST2214
   170 DO 390 I=IMIN,IMAX                                                       FAST2215
       IF(M.LE.2)GO TO 190                                                      FAST2216
       IF((NXT - I)*(IP - I).EQ.0)GO TO 390                                     FAST2217
       DO 180 J=1,NBMAX                                                         FAST2218
       IF(NS(J,I).EQ.0)GO TO 390                                                FAST2219
       IF(IABS(NSP - NS(J,I)/1000).EQ.1000)GO TO 190                            FAST2220
   180 CONTINUE                                                                 FAST2221
       GO TO 390                                                                FAST2222
   190 DO 340 J=1,NBMAX                                                         FAST2223
       IF(NS(J,I).EQ.0)GO TO 350                                                FAST2224
       KK = NS(J,I)/1000000                                                     FAST2225
       K = NS(J,I)/1000 - 1000*KK                                               FAST2226
       IF((M.GT.1).AND.(K.EQ.LL))GO TO 340                                      FAST2227
       KK = 2*KK - 1                                                            FAST2228
       IF(ND(K).GT.0)GO TO 210                                                  FAST2229
       ND(K) = 1                                                                FAST2230
       U(K) = AZ(K)                                                             FAST2231
       V(K) = 0.0                                                               FAST2232
       W(K) = 0.0                                                               FAST2233
       MAX = NTP(K)                                                             FAST2234
       DO 200 L=1,MAX                                                           FAST2235
       U(K) = U(K) + X(L)*A(L,K)                                                FAST2236
       V(K) = V(K) + C(L)*A(L,K)                                                FAST2237
       IF(L.GT.3)W(K) = W(K) + CX(L)*A(L,K)                                     FAST2239
   200 CONTINUE                                                                 FAST2240
   210 IF(M.LE.1)GO TO 220                                                      FAST2241
       IF(FLOAT(KK)*(U(K) + STT*(V(K) + STT*W(K)))) 340,340,390                 FAST2242
   220 IF(ND(K) - 2) 230,290,340                                                FAST2243
   230 ND(K) = 2                                                                FAST2244
       IF(W(K)) 250,240,260                                                     FAST2245
   240 IF(V(K).EQ.0.0)GO TO 310                                                 FAST2246
       SI(1,K) = -U(K)/V(K)                                                     FAST2247
       IF(SI(1,K).LT.0.0)GO TO 310                                              FAST2248
```

```
      SI(2,K) = SI(1,K)                                                 FAST2249
      GO TO 290                                                         FAST2250
  250 L = 2                                                             FAST2251
      GO TO 270                                                         FAST2252
  260 L = 1                                                             FAST2253
  270 E = -0.5*V(K)/W(K)                                                FAST2254
      F = U(K)/W(K)                                                     FAST2255
      H = E**2 - F                                                      FAST2256
      IF(F.LE.0.0)GO TO 280                                             FAST2257
      IF(E.LE.0.0)GO TO 310                                             FAST2258
      IF(H.LE.0.0)GO TO 310                                             FAST2259
  280 H = SQRT(H)                                                       FAST2260
      SI(L,K) = E - H                                                   FAST2261
      L = L + 1 - 2*(L/2)                                               FAST2262
      SI(L,K) = E + H                                                   FAST2263
  290 L = 1                                                             FAST2264
      IF(KK.GT.0)L = 2                                                  FAST2265
      SB = SI(L,K) - STT                                                FAST2266
      IF(SB) 300,320,330                                                FAST2267
  300 L = L + 1 - 2*(L/2)                                               FAST2268
      IF(SI(L,K).GT.STT)GO TO 340                                       FAST2269
  310 ND(K) = 3                                                         FAST2270
      GO TO 340                                                         FAST2271
  320 IF(FLOAT(KK)*(V(K) + 2.0*STT*W(K)).LT.0.0)GO TO 340               FAST2272
  330 IF(SB.GE.SP)GO TO 340                                             FAST2273
      SP = SB                                                           FAST2274
      JJ = J                                                            FAST2275
      LL = K                                                            FAST2276
  340 CONTINUE                                                          FAST2277
  350 IF(M.LE.1)GO TO 360                                               FAST2278
      II = I                                                            FAST2279
      NS(JJ,IP) = 1000*NSP + I                                          FAST2280
      GO TO 410                                                         FAST2281
  360 IF(JJ.EQ.0)GO TO 415                                              FAST2282
  370 NRG(N) = I                                                        FAST2283
      IF(STM.GT.(STT + SP))GO TO 380                                    FAST2284
      NSC(N) = 0                                                        FAST2285
      ST(N) = STM - STT                                                 FAST2286
      GO TO 420                                                         FAST2287
  380 NSC(N) = LL                                                       FAST2238
      ST(N) = SP                                                        FAST2288
      STT = STT + SP                                                    FAST2289
      IF(NTP(LL).LE.3)ND(LL) = 3                                        FAST2290
      NSP = NS(JJ,I)/1000                                               FAST2291
      NXT = NS(JJ,I) - 1000*NSP                                         FAST2292
      NXXX = 0                                                          FAST2293
      IF(NXT.LE.NRMAX) GO TO 385                                        FAST2294
```



```
      NXXX = NXT - NRMAX                                                FAST2295
      IF(NXXX.GT.10) GO TO 420                                          FAST2296
      NXT = 0                                                           FAST2297
  385 IP = I                                                            FAST2298
  390 CONTINUE                                                          FAST2299
  400 CONTINUE                                                          FAST2300
      NS(JJ,IP) = 1000*NSP + MIN0(999,NRMAX*NXXX+1)                     FAST2301
      GO TO 420                                                         FAST2302
  410 CONTINUE                                                          FAST2303
      N = NSTMAX + 1                                                    FAST2304
  415 N = N - 1                                                         FAST2305
  420 NST = N                                                           FAST2306
  430 RETURN                                                            FAST2307
      END                                                               FAST2308
$IBFTC NORMLE  M94/2,XR7                                                FAST2309
CNORMAL*SURFACE NORMAL CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*66FAST2310
      SUBROUTINE NORMAL(NNN,SSS,XXX,CCC,DDD)                            FAST2311
      COMMON        H(1)                                                FAST2312
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST2313
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST2314
     2              NUNITX,NIMAX ,NMPAX ,NORDER,NOOWN ,INELAS,NTRANS,   FAST2315
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2316
     4              NDMAX ,NGMOD ,MOPENT,NDMOD ,NPDMAX,NPCMOD,NSDMAX,   FAST2317
     5              NVDMAX,NVDMCD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2318
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST2319
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST2320
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST2321
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST2322
     3              INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,   FAST2323
     4              IVOL  ,ICOT  ,IXCT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST2324
     5              IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST2325
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST2326
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST2327
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST2328
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST2329
     1              IFXT  ,IFXE  ,IFXA                                  FAST2330
      CALL NOROUM(NAMAX,I,H(INTP),H(IA),NNN,SSS,XXX,CCC,DDD)            FAST2331
      RETURN                                                            FAST2332
      END                                                               FAST2333
$IBFTC NOROM1  M94/2,XR7                                                FAST2334
CNOROUM*SURFACE NORMAL CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WANL*66FAST2335
      SUBROUTINE NOROUM(L1,L2,NTP,A,NNN,STT,XP,CP,C)                    FAST2336
      DIMENSION NTP(1),A(L1,L2),XP(3),CP(3),X(3),C(3)                   FAST2337
      DO 10 I=1,3                                                       FAST2338
   10 X(I) = XP(I) + STT*CP(I)                                          FAST2339
      I = NNN                                                           FAST2340
      FST = 0.0                                                         FAST2341
```

```
      L = 2
      MAX = NTP(I)
      DO 20 J=1,3
      C(J) = 0.0
      K = L
      L = K + 1 - 3*(K/3)
      IF((L+6).LE.MAX)C(J) = X(L)*A(L+6,I)
      IF((J+6).LE.MAX)C(J) = C(J) + X(K)*A(J+6,I)
      IF((J+3).LE.MAX)C(J) = C(J) + 2.0*X(J)*A(J+3,I)
      IF(J.LE.MAX)C(J) = C(J) + A(J,I)
   20 FST = FST + C(J)**2
      IF(FST.NE.1.0)FST = SQRT(FST)
      DO 30 J=1,3
   30 C(J) = C(J)/FST
      RETURN
      END
$IBFTC ROTAT    M94/2,XR7
CROTATE*ROTATION MATRIX
      SUBROUTINE ROTATE(CPH,SPH,CTH,STH,ROT)
      DIMENSION ROT(3,3)
   10 ROT(1,1) = CTH*CPH
      ROT(2,1) = -STH
      ROT(3,1) = CTH*SPH
      ROT(1,2) = STH*CPH
      ROT(2,2) = CTH
      ROT(3,2) = STH*SPH
      ROT(1,3) = -SPH
      ROT(2,3) = 0.0
      ROT(3,3) = CPH
      RETURN
      END
$IBFTC CROT     M94/2,XR7
CROTATC*ROTATION MATRIX GIVEN DIRECTION VECTOR
      SUBROUTINE ROTATC(C,ROT)
      DIMENSION C(3)
      CPH = C(3)
      SPH = SQRT(AMAX1(0.0,1.0 - CPH**2))
      IF(SPH.GT.0.0)GO TO 10
      CTH = 1.0
      STH = 0.0
      GO TO 20
   10 CTH = C(1)/SPH
      STH = C(2)/SPH
   20 CALL ROTATE(CPH,SPH,CTH,STH,ROT)
      RETURN
      END
$IBFTC DOT      M94/2,XR7
```



```
CCOSINE*DOT PRODUCT
      FUNCTION COSINE(X,Y)
      DIMENSION X(3),Y(3)
      A = 0.0
      DO 10 I=1,3
   10 A = A + X(I)*Y(I)
      COSINE = A
      RETURN
      END
$IBFTC TALLYS  M94/2,XR7
CDETECT*FLUX CALCULATIONS FOR PROGRAM FASTER*T.M.JORDAN*WANL*1966
      SUBROUTINE DETECT(III,JJJ,AAA,BBB,CCC)
      COMMON        H(1)
      COMMON/DUMDET/I      ,M     ,FST   ,ANG   ,STV
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
     4              NOMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,
     6              NPRINT,NUNITS,NUMBER,KALIDE
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,INTL  ,IRHO  ,IXR   ,IELL  ,
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,
     3              INTG  ,IELF  ,IFGW  ,ITDS  ,IIDV  ,IIDR  ,IIDS  ,
     4              IVOL  ,ICOT  ,IXDT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,
     5              IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,
     1              IFXT  ,IFXE  ,IFXA
      I = III
      M = JJJ
      FST = AAA
      ANG = BBB
      STV = CCC
      IP = NEXT
      IT = IP+MAX0(NLMAX,2)
      CALL DETDUM(1,NGMAX,NEMOD,MOMENT,H(IWS),H(IES),H(INTG),H(IFXT),
     1H(IFXP),H(IFXE),H(IFXA),H(IRHO),H(IAE),H(IBE),H(ISSH),H(ISGT),
     2H(IXMP),H(IGIM),H(IP),H(IT),H(IAIM))
      RETURN
      END
$IBFTC TALLY   M94/2,XR7
CDETECT*FLUX TALLYS FOR PROGRAM FASTER
      SUBROUTINE DETDUM(L1,L2,L3,L4,WS,ES,NTG,FXT,FXP,FXE,FXA,RHO,AE,BE,
```

Astronuclear Laboratory

```
     1SSH,SGT,XMP,GIM,P,T,AIM)                                          FAST2436
      DIMENSION WS(1),ES(1),NTG(1),FXT(L2,L1),FXP(L2,L1),FXE(L2,L1),   FAST2437
     1FXA(L4,L2,L1),RHO(1),AE(1),BE(1),SSH(1),SGT(L3,L1),XMP(1),GIM(1), FAST2438
     2P(1),T(1),AIM(1)                                                  FAST2439
      COMMON/DUMDET/I      ,M      ,FST    ,ANG    ,STV                 FAST2440
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BOC(3),ATA   ,ATB    ,ATC    , FAST2441
     1              ATO    ,AT     ,BT     ,AS     ,BS     ,JMIN  ,JMAX  , FAST2442
     2              KMIN   ,KMAX   ,JBAR   ,NZERO                       FAST2443
      COMMON/FLUXES/KKK    ,NTALLY,ERROR ,TOTALN,TOTALE,RHON  ,SNORM ,TOTFAST2444
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,  FAST2445
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,  FAST2446
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NODWN ,INELAS,NTRANS,  FAST2447
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,  FAST2448
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,  FAST2449
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,  FAST2450
     6              NPRINT,NUNITS,NUMBER,KALIDE                        FAST2451
      COMMON/POINTX/NTOTAL,III     ,MOM                                FAST2452
      COMMON/REGSCA/NSRMAX,INSR                                        FAST2453
C ANGULAR MOMENTS                                                      FAST2454
      IF(NLMAX.LE.0)GO TO 20                                           FAST2455
      P(1) = ANG                                                       FAST2456
      P(2) = 1.5*ANG*ANG-0.5                                           FAST2457
      IF(NLMAX.LE.2)GO TO 20                                           FAST2458
      DO 10 L=3,NLMAX                                                  FAST2459
   10 P(L) = (FLOAT(2*L-1)*ANG*P(L-1) - FLOAT(L-1)*P(L-2))/FLOAT(L)    FAST2460
   20 IF(NTMAX.LE.0)GO TO 30                                           FAST2461
      IF(M.GE.0)GO TO 30                                               FAST2462
      T(1) = TOT/SNORM                                                 FAST2463
      IF(NTMAX.EQ.1)GO TO 30                                           FAST2464
      DO 25 L=2,NTMAX                                                  FAST2465
   25 T(L) = T(1)*T(L-1)                                               FAST2466
   30 DO 80 K=JMIN,JMAX                                                FAST2467
      IF(WS(K).EQ.0.0                   )GO TO 80                      FAST2468
      GST = FST*WS(K)*EXP(-XMP(K))                                     FAST2469
      IF(M.LT.0)GC TO 40                                               FAST2470
      SIG = RHON   *(AE(K)*SSH(K) + BE(K)*SSH(K+1))                    FAST2471
      IF(M.GT.0)SIG = SIG + (AE(K)*SGT(K,M) + BE(K)*SGT(K+1,M))        FAST2472
      ATN = EXP(-SIG*STV)                                              FAST2473
      AST = (1.0 - ATN)/SIG                                            FAST2474
      GST = GST*AST                                                    FAST2475
      XMP(K) = XMP(K) + SIG*STV                                        FAST2476
   40 IF(GST.EQ.0.0)GO TO 80                                           FAST2477
      J = NTG(K)                                                       FAST2478
      FXT(J,I) = FXT(J,I) + GST                                        FAST2479
      FXP(J,I) = FXP(J,I) + GST                                        FAST2480
      FXE(J,I) = FXE(J,I) + GST*ES(K)                                  FAST2481
      HST = GST*XMP(K)*GIM(K)*AIM(K)                                   FAST2482
```

Astronuclear Laboratory


```
      HST = ABS(HST)                                                   FAST2483
      IF(ABS(GST).GT.(ERROR*FXP(J,I))) NTALLY = NTALLY + 1             FAST2484
      TOTALN = TOTALN + HST                                            FAST2485
      TOTALE = TOTALE + HST*ES(K)                                      FAST2486
      FXA(1,J,I) = FXA(1,J,I) + GST                                    FAST2487
      KK = MOM + 1                                                     FAST2488
      IF(KK.GT.1) FXA(KK,J,I) = FXA(KK,J,I) + GST                      FAST2489
      LL = NVMOD + 1                                                   FAST2490
      KK = LL + INSR                                                   FAST2491
      IF(INSR.GT.0) FXA(KK,J,I) = FXA(KK,J,I) + GST                    FAST2492
      LL = LL + NSRMAX                                                 FAST2493
      KK = LL + KKK                                                    FAST2494
      IF(KKK.LE.NCMAX) FXA(KK,J,I) = FXA(KK,J,I) + GST                 FAST2495
      KK = LL + NCMAX                                                  FAST2496
      IF(NLMAX.LE.0)GO TO 60                                           FAST2497
      DO 50 L=1,NLMAX                                                  FAST2498
      KK = KK + 1                                                      FAST2499
   50 FXA(KK,J,I) = FXA(KK,J,I) + GST*P(L)                             FAST2500
   60 IF(NTMAX.LE.0)GO TO 80                                           FAST2501
      IF(M.LT.0)GC TO 68                                               FAST2502
      AST = SIG - STV*ATN/(1.0 - ATN)                                  FAST2503
      T(1) = (TOT + AST)/SNORM                                         FAST2504
      IF(NTMAX.EQ.1)GO TO 68                                           FAST2505
      DO 63 L=2,NTMAX                                                  FAST2506
   63 T(L) = T(1)*T(L-1)                                               FAST2507
   68 CONTINUE                                                         FAST2508
      DO 70 L=1,NTMAX                                                  FAST2509
      KK = KK + 1                                                      FAST2510
   70 FXA(KK,J,I) = FXA(KK,J,I) + GST*T(L)                             FAST2511
   80 CONTINUE                                                         FAST2512
      RETURN                                                           FAST2513
      END                                                              FAST2514
$IBFTC ANSWRE  M94/2,XR7                                               FAST2515
CANSWER*FLUX PRINTOUT FCR PROGRAM FASTER                               FAST2516
      SUBROUTINE ANSWER(NNN)                                           FAST2517
      COMMON        H(1)                                               FAST2518
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,  FAST2519
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,  FAST2520
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NODWN ,INELAS,NTRANS,  FAST2521
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,  FAST2522
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,  FAST2523
     5              NVDMAX,NVDMOD,NPCINT,MODELP,MODELQ,MODELU,MODELV,  FAST2524
     6              NPRINT,NUNITS,NUMBER,KALIDE                        FAST2525
      COMMON/INDEXS/INTP   ,IAZ    ,IISV   ,IMTL   ,IRHO   ,IXR    ,IELL   , FAST2526
     1              IELW   ,IAE    ,IBE    ,IMSG   ,INPC   ,IJSN   ,IJSX   , FAST2527
     2              IXTR   ,ISUV   ,IATN   ,IATW   ,IESB   ,ISSH   ,IDEN   , FAST2528
     3              INTG   ,IELF   ,IFGW   ,ITDS   ,IIDV   ,IIDR   ,IIDS   , FAST2529
```

Astronuclear Laboratory

```
     4                 IVOL  ,ICDT  ,IXDT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,      FAST2530
     5                 IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,      FAST2531
     6                 ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,      FAST2532
     7                 ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,INC   ,      FAST2533
     8                 IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,      FAST2534
     9                 INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,      FAST2535
     1                 IFXT  ,IFXE  ,IFXA                                     FAST2536
      IDO = NEXT                                                              FAST2537
      ILE = NEXT + MOMENT                                                     FAST2538
      CALL ANSDUM(NGMAX,1,AGMDC,MOMENT,H(IDO),H(IIDV),H(ILE),H(IFXT),         FAST2539
     1            H(IFXE),H(IFXS),H(IFXA),H(IELF),H(IFGW),H(IWS),H(IES),FAST2540
     2H(INC),H(IEC),H(ITDS),H(IRSP),NNN,H(INS),NBMAX)                         FAST2541
      RETURN                                                                  FAST2542
      END                                                                     FAST2543
$IBFTC ANSDM1  M94/2,XR7                                                      FAST2544
CANSDUM*FINAL PROCESSING AND PRINTOUT OF FASTER FLUXES*T.M.JORDAN*WANL**FAST2545
      SUBROUTINE ANSDUM(L1,L2,L3,L4,IDO,IDV,TLE,FXT,FXE,FXS,FXA,ELF,FGW,FAST2546
     1AE,BE,ERR,FLX,TDS,RSP,NNN,NS,L5)                                        FAST2547
      COMMON NOX(1)                                                           FAST2548
      COMMON/REGSCA/NSRMAX,INSR                                               FAST2549
      DIMENSION LIM(6),IDO(1),IDV(1),TLE(2,1),TYPE(2,5),OUT(8),               FAST2550
     1          FXT(L1,L2),FXE(L1,L2),FXS(L1,L2),FGW(1),AE(1),BE(1),          FAST2551
     2          FLX(1),ERR(1),TDS(3,1),OUS(8),EMN(8),EMX(8),RSP(L3,L2),       FAST2552
     3          ELF(1),FXA(L4,L1,L2)                                          FAST2553
      EQUIVALENCE (LIM(1),LIM1),(LIM(2),LIM2),(LIM(3),LIM3),(LIM(4),          FAST2554
     1LIM4),(LIM(5),LIM5),(LIM(6),NMMOD),(OUT(1),OUT1),(OUT(2),OUT2),         FAST2555
     2             (OUT(3),OUT3),(OUT(4),OUT4),(OUT(7),OUT7),                 FAST2556
     3             (OUT(8),OUT8)                                              FAST2557
      DATA TYPE/4H  SO,4HURCE,4H  RE,4HGION,4H SCA,4HTTER,4H ANG,4HULAR,FAST2558
     14H SPA,4HTIAL/                                                          FAST2559
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,         FAST2560
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,         FAST2561
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,         FAST2562
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,         FAST2563
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPOMAX,NPOMOD,NSOMAX,         FAST2564
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,         FAST2565
     6              NPRINT,NUNITS,NUMBER,KALIDE                               FAST2566
      COMMON/POINTX/NTOTAL,III   ,MOM                                         FAST2567
      COMMON/TAPEID/M1    ,M2                                                 FAST2568
      COMMON/CASEID/KASE  ,NPAGE ,LINES ,LINEX ,TITLEC(36)                    FAST2569
      COMMON/ENDRUN/NPDRUN                                                    FAST2570
      DIMENSION IANS(8),MPNR(8),NS(L5,L2)                                     FAST2571
      DATA THIS,PASS/4H ITE,4HRANT/,RIGHT/3H  (/                              FAST2572
      IF(NNN.EQ.1) MTOTAL = 0                                                 FAST2573
      LIM1 = 1                                                                FAST2574
      IDO(1) = NNN                                                            FAST2575
      TLE(1,1) = THIS                                                         FAST2576
```



```
      TLE(2,1) = PASS                                                         FAST2577
      LIM2 = LIM1 + NVMOD                                                     FAST2578
      LIM3 = LIM2 + NSRMAX                                                    FAST2579
      LIM4 = LIM3 + NCMAX                                                     FAST2580
      LIM5 = LIM4 + NLMAX                                                     FAST2581
      NMMOD = LIM5 + NTMAX                                                    FAST2582
      DO 60 I=1,5                                                             FAST2583
      IF(LIM(I).EQ.LIM(I+1)) GO TO 60                                         FAST2584
      MIN = LIM(I) + 1                                                        FAST2585
      MAX = LIM(I+1)                                                          FAST2586
      IF(I - 2) 5,15,20                                                       FAST2587
    5 CONTINUE                                                                FAST2588
      DO 10 J=MIN,MAX                                                         FAST2589
   10 IDO(J) = IDV(J-1)                                                       FAST2590
      GO TO 40                                                                FAST2591
   15 K = 0                                                                   FAST2592
      DO 18 J=MIN,MAX                                                         FAST2593
      K = K + 1                                                               FAST2594
   18 IDO(J) = NOX(K)                                                         FAST2595
      GO TO 40                                                                FAST2596
   20 K = I/4                                                                 FAST2597
      DO 30 J=MIN,MAX                                                         FAST2598
   30 IDO(J) = J + K - MIN                                                    FAST2599
   40 DO 50 J=MIN,MAX                                                         FAST2600
      DO 50 K=1,2                                                             FAST2601
   50 TLE(K,J) = TYPE(K,I)                                                    FAST2602
   60 CONTINUE                                                                FAST2603
   70 NPF = NNN*NUNITS                                                        FAST2604
      IF(MODELP.EQ.0) NTOTAL = NPF                                            FAST2605
      XN = FLOAT(NTOTAL)                                                      FAST2606
      XTI = XN/FLOAT(NTOTAL - MTOTAL)                                         FAST2607
      MTOTAL = NTOTAL                                                         FAST2608
      XS = XN**2                                                              FAST2609
      DO 240 I=1,NOMOD                                                        FAST2610
      IOUT = I                                                                FAST2611
      IF(NPOINT.EQ.1) IOUT = III                                              FAST2612
      IF(LABEL(4).GE.0) WRITE(M2,2000)IOUT,NPF                                FAST2613
 2000 FORMAT(1X,32(1H*),19HFLUXES FOR DETECTOR,I4,6H AFTER,I6,8H PACKETSFAST2614
     1,32(1H*)/14X,22HCALCULATED     AVERAGE,3(24H NUMBER FLUX ENERGY FLFAST2615
     2UX)/15X,21HPRECISION  ENERGY-MEV,2(12H  THIS GROUP),2(12H  DERIVATFAST2616
     3IVE),2(12H  CUMULATIVE))                                                FAST2617
      LINES = LINES - 1                                                       FAST2618
      OUT7 = 0.0                                                              FAST2619
      OUT8 = 0.0                                                              FAST2620
      DO 110 J=1,NGMAX                                                        FAST2621
      DO 80 K=1,6                                                             FAST2622
   80 OUT(K) = 0.0                                                            FAST2623
```

```
      OUT3 = FXT(J,I)/XN
      IF(OUT3.EQ.0.0)GO TO 100
      OUT4 = FXE(J,I)/XN
      OUT2 = OUT4/OUT3
      DO 90 K=5,6
      OUT(K) = OUT(K-2)/FGW(J)
   90 OUT(K+2) = OUT(K+2) + OUT(K-2)
      FST = (FXS(J,I) - XN*OUT3**2)/XS
      IF(FST.GT.0.0)OUT1 = SQRT(FST)/OUT3
  100 IF(LABEL(1).GE.0)WRITE(M2,2010)J,OUT
 2010 FORMAT(1X,5HGROUP,I4,3H...,1PE11.4,7E12.4)
      AE(J) = (OUT2 - ELF(J+1))/FGW(J)
      BE(J) = (ELF(J) - OUT2)/FGW(J)
      FLX(J) = OUT3
  110 ERR(J) = OUT1
      IF(NFMAX.EQ.0)GO TO 170
      DO 160 J=1,NFMAX,8
      MAX = MINO(J+7,NFMAX)
      MOD = MAX - J + 1
      IF(LABEL(3).GE.0)WRITE(M2,2020)IOUT,NPF,((TDS(L,K),L=1,3),K=J,MAX)
 2020 FORMAT(1X,24(1H*),34HNUMBER FLUX RESPONSES FOR DETECTOR,I4,6H AFTE
     1R,I6,8H PACKETS,25(1H*)/12X,24A4)
      LINES = LINES - 1
      DO 120 K=1,MOD
      OUS(K) = 0.0
      EMN(K) = 0.0
  120 EMX(K) = 0.0
      DO 140 K=1,NGMAX
      L = 0
      DO 130 M=J,MAX
      L = L + 1
      FST = FLX(K)*(AE(K)*RSP(K,M) + BE(K)*RSP(K+1,M))
      OUT(L) = FST
      OUS(L) = OUS(L) + FST
      FST = FST*ERR(K)
      EMN(L) = EMN(L) + FST**2
  130 EMX(L) = EMX(L) + FST
      IF(LABEL(1).GE.0)WRITE(M2,2010)K,(OUT(L),L=1,MOD)
  140 CONTINUE
      IF(LABEL(1).GE.0)WRITE(M2,2030)(OUS(L),L=1,MOD)
 2030 FORMAT(1X,6HTOTALS,6(1H.),1PE11.4,7E12.4)
      DO 150 L=1,MOD
      OUT(L) = 0.0
      FST = OUS(L)
      OUS(L) = 0.0
      IF(FST.EQ.0.0)GO TO 150
      OUT(L) = SQRT(EMN(L))/FST
```



```
      OUS(L) = EMX(L)/FST
  150 CONTINUE
      IF(LABEL(1).GE.0)WRITE(M2,2035)(OUT(L),L=1,MOD)
 2035 FORMAT(1X,12HMIN ERROR...,1PE11.4,7E12.4)
      IF(LABEL(1).GE.0)WRITE(M2,2040)(OUS(L),L=1,MOD)
 2040 FORMAT(1X,12HMAX ERROR...,1PE11.4,7E12.4)
  160 CONTINUE
  170 IF(NMMOD.EQ.0)GO TO 240
      DO 230 J=1,NMMOD,8
      MAX = MINO(J+7,NMMOD)
      MOD = MAX - J + 1
      IF(LABEL(3).GE.0)WRITE(M2,2050)IOUT,NPF,((TLE(K,L),K=1,2),IDO(L),L
     1=J,MAX)
 2050 FORMAT(1X,25(1H*),32HNUMBER FLUX MOMENTS FOR DETECTOR,I4,6H AFTER,
     1I6,8H PACKETS,26(1H*)/12X,8(1X,2A4,I3))
      LINES = LINES - 1
      DO 190 K=1,NGMAX
      L = 0
      DO 180 M=J,MAX
      L = L + 1
  180 OUT(L) = FXA(M,K,I)/XN
      IF(J.EQ.1) OUT1 = XTI*OUT1
      IF(LABEL(1).GE.0)WRITE(M2,2010)K,(OUT(L),L=1,MOD)
  190 CONTINUE
      IF(NFMAX.EQ.0)GO TO 230
      DO 220 K=1,NFMAX
      DO 200 L=1,MOD
  200 OUT(L) = 0.0
      DO 210 L=1,NGMAX
      M = 0
      DO 210 N=J,MAX
      M = M + 1
  210 OUT(M) = OUT(M) + FXA(N,L,I)*(AE(L)*RSP(L,K)+BE(L)*RSP(L+1,K))/XN
      IF(J.EQ.1) OUT1 = XTI*OUT1
      IF(LABEL(1).GE.0)WRITE(M2,2060)(TDS(L,K),L=1,3),(OUT(L),L=1,MOD)
 2060 FORMAT(1X,3A4,1PE11.4,7E12.4)
  220 CONTINUE
  230 CONTINUE
  240 CONTINUE
      DO 250 I=1,NDMOD
      DO 250 J=1,NGMAX
  250 FXA(I,J,I) = 0.0
      IF(NNN.LT.NPRINT)GO TO 900
      IF(NPOINT.EQ.0) GO TO 300
      IF(NPORUN.LT.NPDMAX) GO TO 870
  300 IF(LABEL(-1).GE.0) WRITE(M2,3000)
 3000 FORMAT(1X,22(1H*),63HBOUNDARY SEARCH PARAMETERS, (SURFACE,MOST PRO
```

```
      1BABLE NEXT REGION),22(1H*))                                      FAST2718
      DO 340 I=1,NRMAX                                                  FAST2719
      DO 330 J=1,NBMAX,8                                                FAST2720
      MAX = MINO(NBMAX,J+7)                                             FAST2721
      K = 0                                                             FAST2722
      DO 310 L=J,MAX                                                    FAST2723
      M = NS(L,I)                                                       FAST2724
      IF(M.EQ.0) GO TO 320                                              FAST2725
      K = K + 1                                                         FAST2726
      N = M/1000                                                        FAST2727
      MPNR(K) = M - 1000*N                                              FAST2728
      M = N/1000                                                        FAST2729
  310 IANS(K) = (2*M - 1)*(N - 1000*M)                                  FAST2730
  320 IF(K.EQ.0)GO TO 340                                               FAST2731
      IF(LABEL(1).GE.0) WRITE(M2,3010)I,(RIGHT,IANS(L),MPNR(L),L=1,K)   FAST2732
 3010 FORMAT(1X,6HREGION,I5,8(A3    ,I4,1H,,I3,1H)))                    FAST2733
  330 CONTINUE                                                          FAST2734
  340 CONTINUE                                                          FAST2735
  870 CONTINUE                                                          FAST2736
      DO 890 I=1,NOMOD                                                  FAST2737
      DO 890 J=1,NGMAX                                                  FAST2738
      FXT(J,I) = 0.0                                                    FAST2739
      FXE(J,I) = 0.0                                                    FAST2740
      FXS(J,I) = 0.0                                                    FAST2741
      IF(MOMENT.EC.0)GO TO 890                                          FAST2742
      DO 880 K=1,MOMENT                                                 FAST2743
  880 FXA(K,J,I) = 0.0                                                  FAST2744
  890 CONTINUE                                                          FAST2745
  900 RETURN                                                            FAST2746
      END                                                               FAST2747
$IBFTC PSTARX  M94/2,XR7                                                FAST2748
CPSTAR *RANDOM SELECTION OF SOURCE VECTOR FOR PROGRAM FASTER*T.M.JORDAN*FAST2749
      FUNCTION PSTAR(NNN,XXX)                                           FAST2750
      DIMENSION XXX(1)                                                  FAST2751
      COMMON        H(1)                                                FAST2752
      COMMON/DUMPST/NN     ,PDT    ,X(3)                                FAST2753
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST2754
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST2755
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST2756
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2757
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST2758
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2759
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST2760
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,INTL  ,IRHO  ,IXR   ,IELL  ,   FAST2761
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST2762
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST2763
     3              INTG  ,IELF  ,IFGW  ,ITOS  ,IIDV  ,IIOR  ,IIDS  ,   FAST2764
```



```
     4              IVOL  ,ICDT  ,IXDT  ,IRSI  ,IALP  ,IVMD  ,IGIM  ,   FAST2765
     5              IAIM  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST2766
     6              ISPW  ,ISPE  ,IATD  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST2767
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST2768
     8              IEC   ,IND   ,IU    ,IV    ,IW    ,ISI   ,IST   ,   FAST2769
     9              INRG  ,INSC  ,ISTP  ,INRP  ,INCP  ,IFXP  ,IFXS  ,   FAST2770
     1              IFXT  ,IFXE  ,IFXA                                  FAST2771
      CALL PSTDUM(NXMAX,5,1,H(IRSI),H(INPC),H(IVEE),H(IVMD),H(IALP),    FAST2772
     1            H(INSG),H(IXTR))                                      FAST2773
      NNN = NN                                                          FAST2774
      DO 10 I=1,3                                                       FAST2775
   10 XXX(I) = X(I)                                                     FAST2776
      PSTAR = PDT                                                       FAST2777
      RETURN                                                            FAST2778
      END                                                               FAST2779
$IBFTC PSTDM1  M94/2,XR7                                                FAST2780
CPSTDUM*RANDOM SELECTION OF SOURCE VECTOR FOR PROGRAM FASTER*T.M.JORDAN*FAST2781
      SUBROUTINE PSTDUM(L1,L2,L3,RSI,NPC,VEE,VMD,ALP,NSG,XTR)           FAST2782
      COMMON/DUMPST/NN     ,PDT    ,X(3)                                FAST2783
      DIMENSION RSI(1),NPC(5,1),VEE(L1,L2,L3),VMD(5,1),ALP(5,1),NSG(1), FAST2784
     1          XTR(3,1)                                                FAST2785
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST2786
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,   FAST2787
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST2788
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2789
     4              NOMAX ,NGMOD ,MOMENT,NOMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST2790
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2791
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST2792
      COMMON/POINTX/NTOTAL,III    ,MOM                                  FAST2793
      COMMON/LLSREG/LLREG                                               FAST2794
      DIMENSION VB(5)                                                   FAST2795
   OLD METHOD, SELECT VOLUME                                            FAST2796
      IF(NVMAX.GT.1)GO TO 10                                            FAST2797
      PDT = 1.0                                                         FAST2798
      N = 1                                                             FAST2799
      GO TO 40                                                          FAST2800
   10 R = RANNO(NMB)                                                    FAST2801
      FST = 0.0                                                         FAST2802
      DO 20 N=1,NVMAX                                                   FAST2803
      FST = FST + RSI(N)                                                FAST2804
      IF(FST.GE.R)GO TO 30                                              FAST2805
   20 CONTINUE                                                          FAST2806
   30 PDT = 1.0/RSI(N)                                                  FAST2807
   40 DO 50 I=1,3                                                       FAST2808
      J = NPC(I,N)                                                      FAST2809
   50 PDT = PDT*SAMPLE(VEE(1,I,N),VEE(J,I,N),VMD(I,N),ALP(I,N),VB(I))   FAST2810
      LLREG = N                                                         FAST2811
```

```
      IF(NSG(N) - 1) 60,80,90
   60 DO 70 I=1,3
   70 X(I) = VB(I)
      GO TO 110
   80 X(3) = VB(3)
      R = VB(1)
      GO TO 100
   90 R = VB(1)*SQRT(1.0 - VB(3)**2)
      X(3) = VB(1)*VB(3)
  100 X(1) = R*COS(VB(2))
      X(2) = R*SIN(VB(2))
      IF(NPC(1,N).GT.1)PDT = PDT*VB(1)**NSG(N)
  110 DO 120 I=1,3
  120 X(I) = X(I) + XTR(I,N)
      NN = LOCATE(NGS,X)
      RETURN
      END
$IBFTC SPHEER  M94/2,XR7
CSPHERE*INITIAL SOURCE VECTOR FROM SPHERE FOR PROGRAM FASTER*T.M.JORDAN*
      FUNCTION SPHERE(YYY,AAA,ZZZ,MMM,ST,NRG,NSC)
      DIMENSION YYY(1),ZZZ(1)
      COMMON          H(1)
      COMMON/DUMSPH/NST    ,PDT    ,STC    ,X(3)   ,C(3)
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPOMAX,NPOMOD,NSDMAX,
     5              NVDMAX,NVDMOD,NPCINT,MODELP,MODELQ,MODELU,MODELV,
     6              NPRINT,NUNITS,NUMBER,KALIDE
      COMMON/INDEXS/INTP   ,IAZ    ,IISV   ,IMTL   ,IRHO   ,IXR    ,IELL   ,
     1              IELW   ,IAE    ,IBE    ,INSG   ,INPC   ,IJSN   ,IJSX   ,
     2              IXTR   ,ISUV   ,IATN   ,IATW   ,IESB   ,ISSH   ,IDEN   ,
     3              INTG   ,IELF   ,IFGW   ,IIDS   ,IIDV   ,IIDR   ,IIDS   ,
     4              IVOL   ,ICDT   ,IXCT   ,IRSI   ,IALP   ,IVMD   ,IGIM   ,
     5              IAIM   ,IALM   ,IALH   ,IA     ,INS    ,IVEE   ,IVAL   ,
     6              ISPW   ,ISPE   ,IATD   ,IRSP   ,ISGT   ,IIDE   ,IIDI   ,
     7              ISGS   ,NEXT   ,IXMP   ,ISGR   ,IWS    ,IES    ,IWC    ,
     8              IEC    ,IND    ,IU     ,IV     ,IW     ,ISI    ,IST    ,
     9              INRG   ,INSC   ,ISTP   ,INRP   ,INCP   ,IFXP   ,IFXS   ,
     1              IFXT   ,IFXE   ,IFXA
      DO 10 I=1,3
   10 X(I) = YYY(I)
      CALL SPHDUM(NEMOD,1,H(IISV),H(ISUV),H(IRHO),H(ISSH),H(IMTL),
     1H(ISGT),H(IND),H(IV),H(ISI),ST,NRG,NSC)
      MMM = NST
      AAA = STC
      DO 20 I=1,3
   20 ZZZ(I) = C(I)
      SPHERE = PDT
      RETURN
      END
$IBFTC SPHDM1  M94/2,XR7
CSPHDUM*INITIAL SOURCE VECTOR FROM SPHERE FOR PROGRAM FASTER*T.M.JORDAN*
      SUBROUTINE SPHDUM(L1,L2,ISV,SUV,RHO,SSH,MTL,SGT,XM,TRA,TRB,ST,NRG,
     1NSC)
      DIMENSION ISV(1),SUV(1),NRG(1),ST(1),SGT(L1,L2),SSH(1),MTL(1),
     1          XM(1),TRA(1),TRB(1),RHO(1)
      DIMENSION ROT(3,3),C(3)
      COMMON/DUMSPH/NST    ,PDT    ,STT    ,X(3)   ,C(3)
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPOMAX,NPOMOD,NSDMAX,
     5              NVDMAX,NVDMOD,NPCINT,MODELP,MODELQ,MODELU,MODELV,
     6              NPRINT,NUNITS,NUMBER,KALIDE
      COMMON/POINTX/NTOTAL,III    ,MOM
      COMMON/INDEXG/JZERO
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA    ,ATB    ,ATC    ,
     1              ATD    ,AT     ,BT     ,AS     ,BS     ,JMIN   ,JMAX   ,
     2              KMIN   ,KMAX   ,JBAR   ,NZERO
      DIMENSION XS(3)
      COMMON/SPASHL/PSORS
      DATA PI/3.141593/
      COMMON/LLSREG/LLREG
      DIMENSION NSC(1)
C NEW METHOD, SETUP SPHERICAL VOLUME
  130 STM = VECTOR(X,XCT,C)
      CALL ROTATC(C,ROT)
      NN = LOCATE(NGS,X  )
      PHIMIN = -1.0
      FST = STM**2 - RADIUS**2
      IF(FST.GE.0.0)PHIMIN = SQRT(FST)/STM
      STM = STM + RADIUS
      ATX = -ATA/(1.0 - PHIMIN)
      DO 165 NTRY=1,100
      NTOTAL = NTOTAL + 1
      PDT = SAMPLE(PHIMIN,1.0,1.0,ATX,D(3))*SAMPLE(-PI,PI,0.0,ATB,THE)
      SNP = SQRT(1.0 - D(3)**2)
      D(1) = SNP*COS(THE)
      D(2) = SNP*SIN(THE)
      DO 150 I=1,3
      C(I) = 0.0
```

```
      DO 150 J=1,3                                                      FAST2906
  150 C(I) = C(I) + D(J)*RCT(J,I)                                       FAST2907
      CALL PATH(NN,STM,X  ,C,NST,ST,NRG,NSC)                            FAST2908
      STT = 0.0                                                         FAST2909
      DO 160 I=1,NST                                                    FAST2910
      J = NRG(I)                                                        FAST2911
      IF(ISV(J).GT.0)GO TO 170                                          FAST2912
  160 STT = STT + ST(I)                                                 FAST2913
  165 CONTINUE                                                          FAST2914
      NST = 0                                                           FAST2915
      GO TO 300                                                         FAST2916
  170 NTT = I                                                           FAST2917
      PTOT = 0.0                                                        FAST2918
      XMPT = 0.0                                                        FAST2919
      STX = STT                                                         FAST2920
      DO 220 I=NTT,NST                                                  FAST2921
      J = NRG(I)                                                        FAST2922
      K = ISV(J)                                                        FAST2923
      L = MTL(J)                                                        FAST2924
      SGR = ATC*RHO(J)*SSH(JZERO)                                       FAST2925
      IF(L.GT.0)SGR = SGR + ATC*SGT(JZERO,L)                            FAST2926
      XMPP = SGR*ST(I)                                                  FAST2927
      IF(K.GT.0)GO TO 180                                               FAST2928
      XM(I) = 0.0                                                       FAST2929
      GO TO 210                                                         FAST2930
  180 CALL ARROW(X,STX + 0.01*ST(I),C,XS)                               FAST2931
      LLREG = K                                                         FAST2932
      CALL SZERO(0,1.0,XS,C)                                            FAST2933
      PDB = PSORS                                                       FAST2934
      CALL ARROW (X,STX + 0.5*ST(I),C,XS)                               FAST2935
      CALL SZERO(0,1.0,XS,C)
      PDH = PSORS                                                       FAST2937
      TRA(I) = PDB*SUV(K)*EXP(-XMPT)                                    FAST2938
      ARGARG = 0.0                                                      FASA2939
      IF((PDB*PDH).GT.0.0) ARGARG = ALOG(PDB/PDH)                       FASB2939
      IF(ST(I).GT.0.0) SGR = SGR + 2.0*ARGARG/ST(I)                     FASC2939
      TRB(I) = SGR                                                      FAST2940
      IF(SGR.NE.0.0)GO TO 190                                           FAST2941
      XM(I) = ST(I)*TRA(I)                                              FAST2942
      GO TO 200                                                         FAST2943
  190 XM(I) = TRA(I)*(1.0 - EXP(-SGR*ST(I)))/SGR                        FAST2944
  200 PTOT = PTOT + XM(I)                                               FAST2945
  210 XMPT = XMPT + XMPP                                                FAST2946
  220 STX = STX + ST(I)                                                 FAST2947
      PDS = PTOT*RANNO(NMP)                                             FAST2948
      DO 230 I=NTT,NST                                                  FAST2949
      IF(XM(I).GT.PDS)GO TO 240                                         FAST2950
```



```
      PDS = PDS - XM(I)                                                 FAST2951
  230 STT = STT + ST(I)                                                 FAST2952
      I= NST
  240 NN = NRG(I)                                                       FAST2953
      LLREG = ISV(NN)                                                   FAST2954
      NST = 0
      IF(LLREG.LE.0) GO TO 300
      NST = I                                                           FAST2955
      SGR = TRB(I)                                                      FAST2956
      IF(SGR.NE.0.0)GO TO 250                                           FAST2957
      ST(I) = PDS/TRA(I)                                                FAST2958
      POT = POT/TRA(I)                                                  FAST2959
      GO TO 260                                                         FAST2960
  250 CONTINUE                                                          FAST2961
      ARG = TRA(I) - PDS*SGR                                            FAST2962
      POT = POT/ARG                                                     FAST2963
      ST(I) = -ALOG(ARG/TRA(I))/SGR                                     FAST2964
  260 STT = STT + ST(I)                                                 FAST2965
      POT = POT*PTOT*STT**2                                             FAST2966
  300 RETURN                                                            FAST2967
      END                                                               FAST2968
$IBFTC ARROWV  M94/2,XR7                                                FAST2969
      SUBROUTINE ARROW(X,S,C,Y)                                         FAST2970
      DIMENSION X(3),C(3),Y(3)                                          FAST2971
      DO 10 I=1,3                                                       FAST2972
   10 Y(I) = X(I) + S*C(I)                                              FAST2973
      RETURN                                                            FAST2974
      END                                                               FAST2975
$IBFTC QSTARX  M94/2,XR7                                                FAST2976
CQSTAR *RANDOM SELECTION OF INITIAL DIRECTION VECTOR FOR FASTER*T.JORDANFAST2977
      FUNCTION QSTAR(NN,X,C)                                            FAST2978
      COMMON         H(1)                                               FAST2979
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST2980
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST2981
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,   FAST2982
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST2983
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST2984
     5              NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,   FAST2985
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST2986
      COMMON/INDEXS/INTP  ,IAZ   ,IISV  ,IMTL  ,IRHO  ,IXR   ,IELL  ,   FAST2987
     1              IELW  ,IAE   ,IBE   ,INSG  ,INPC  ,IJSN  ,IJSX  ,   FAST2988
     2              IXTR  ,ISUV  ,IATN  ,IATW  ,IESB  ,ISSH  ,IDEN  ,   FAST2989
     3              INTG  ,IELF  ,IFGW  ,ITOS  ,IIDV  ,IIDR  ,IIDS  ,   FAST2990
     4              IVOL  ,ICDT  ,IXCT  ,IRSI  ,IALP  ,IVMO  ,IGIM  ,   FAST2991
     5              IAIN  ,IALM  ,IALH  ,IA    ,INS   ,IVEE  ,IVAL  ,   FAST2992
     6              ISPW  ,ISPE  ,IATO  ,IRSP  ,ISGT  ,IIDE  ,IIDI  ,   FAST2993
     7              ISGS  ,NEXT  ,IXMP  ,ISGR  ,IWS   ,IES   ,IWC   ,   FAST2994
```

```
     8              IEC    ,IND    ,IU     ,IV     ,IW     ,ISI    ,IST    ,     FAST2995
     9              INRG   ,INSC   ,ISTP   ,INRP   ,INCP   ,IFXP   ,IFXS   ,     FAST2996
     1              IFXT   ,IFXE   ,IFXA                                         FAST2997
      QSTAR = QSTDUM(NN,X,C,AXMAX,5,1,H(ISV),H(NPC),H(IVEE),H(IVMD),            FAST2998
     1              H(IALP),H(XTR),H(INSG))                                      FAST2999
      RETURN                                                                     FAST3000
      END                                                                        FAST3001
$IBFTC QSTDM1  M94/2,XR7                                                         FAST3002
CQSTDUM*RANDOM SELECTION OF INITIAL DIRECTION VECTOR FOR FASTER*T.JORDANFAST3003
      FUNCTION QSTDUM(NN,X,C,L1,L2,L3,ISV,NPC,VEE,VMD,ALP,XTR,NSG)               FAST3004
      DIMENSION X(3),C(3)                                                        FAST3005
      DIMENSION ISV(1),NPC(5,1),VEE(L1,L2,L3),VMD(5,1),ALP(5,1),                 FAST3006
     1          XTR(3,1),NSG(1)                                                  FAST3007
      DIMENSION VB(5),D(3),Z(3),ROT(3,3)                                         FAST3008
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA    ,ATB    ,ATC    ,         FAST3009
     1              ATD    ,AT     ,BT     ,AS     ,BS     ,JMIN   ,JMAX   ,     FAST3010
     2              KMIN   ,KMAX   ,JBAR   ,NZERO                                FAST3011
      COMMON/LLSREG/LLREG                                                        FAST3012
      N = ISV(NN)                                                                FAST3013
      N = LLREG                                                                  FAST3014
      PDA = 1.0                                                                  FAST3015
      DO 10 I=4,5                                                                FAST3016
      J = NPC(I,N)                                                               FAST3017
   10 PDA = PDA*SAMPLE(VEE(1,I,N),VEE(J,I,N),VMD(I,N),ALP(I,N),VB(I))            FAST3018
      D(3) = VB(5)                                                               FAST3019
      SPH = SQRT(1.0 - D(3)**2)                                                  FAST3020
      D(1) = SPH*COS(VB(4))                                                      FAST3021
      D(2) = SPH*SIN(VB(4))                                                      FAST3022
      IF(NSG(N).GT.0)GO TO 30                                                    FAST3023
      DO 20 I=1,3                                                                FAST3024
   20 C(I) = D(I)                                                                FAST3025
      GO TO 90                                                                   FAST3026
   30 DO 40 I=1,3                                                                FAST3027
   40 Z(I) = X(I) - XTR(I,N)                                                     FAST3028
      MAX = NSG(N) + 1                                                           FAST3029
      ARE = 0.0                                                                  FAST3030
      DO 50 I=1,MAX                                                              FAST3031
   50 ARE = ARE + Z(I)**2                                                        FAST3032
      ARE = SQRT(ARE)                                                            FAST3033
      IF(MAX.EQ.3)GO TO 60                                                       FAST3034
      CPH = 1.0                                                                  FAST3035
      SPH = 0.0                                                                  FAST3036
      GO TO 70                                                                   FAST3037
   60 CPH = Z(3)/ARE                                                             FAST3038
      SPH = SQRT(1.0 - CPH**2)                                                   FAST3039
      ARE = SPH*ARE                                                              FAST3040
   70 CTH = Z(1)/ARE                                                             FAST3041
```



```
      STH = Z(2)/ARE                                                             FAST3042
      CALL ROTATE(CPH,SPH,CTH,STH,ROT)                                           FAST3043
      DO 80 I=1,3                                                                FAST3044
      C(I) = 0.0                                                                 FAST3045
      DO 80 J=1,3                                                                FAST3046
   80 C(I) = C(I) + D(J)*ROT(J,I)                                                FAST3047
   90 QSTDUM = PDA                                                               FAST3048
      RETURN                                                                     FAST3049
      END                                                                        FAST3050
$IBFTC USTARX  M94/2,XR7                                                         FAST3051
CUSTAR $RANDOM SELECTION OF SCATTERING POINT FOR PROGRAM FASTER*T.JORDANFAST3052
      FUNCTION USTAR(TTT,XXX,YYY,ZZZ,MMM,ST,NRG)                                 FAST3053
      DIMENSION ZZZ(1),YYY(1),XXX(1),CP(3)                                       FAST3054
      COMMON        H(1)                                                         FAST3055
      COMMON/DUMUST/STZ    ,STS    ,PSI    ,STC    ,NST    ,PDT                  FAST3056
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,            FAST3057
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,            FAST3058
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,            FAST3059
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,            FAST3060
     4              NDMAX ,NGMOD ,MOMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,            FAST3061
     5              NVDMAX,NVDMOD,NPCINT,MODELP,MODELQ,MODELU,MODELV,            FAST3062
     6              NPRINT,NUNITS,NUMBER,KALIDE                                  FAST3063
      COMMON/INDEXS/INTP   ,IAZ    ,IISV   ,IMTL   ,IRHO   ,IXR    ,IELL   ,     FAST3064
     1              IELW   ,IAE    ,IBE    ,INSG   ,INPC   ,IJSN   ,IJSX   ,     FAST3065
     2              IXTR   ,ISUV   ,IATN   ,IATW   ,IESB   ,ISSH   ,IDEN   ,     FAST3066
     3              INTG   ,IELF   ,IFGW   ,ITDS   ,IIDV   ,IIDR   ,IIDS   ,     FAST3067
     4              IVOL   ,ICCT   ,IXGT   ,IRSI   ,IALP   ,IVMD   ,IGIM   ,     FAST3068
     5              IAIM   ,IALM   ,IALH   ,IA     ,INS    ,IVEE   ,IVAL   ,     FAST3069
     6              ISPW   ,ISPE   ,IATO   ,IRSP   ,ISGT   ,IIDE   ,IIDI   ,     FAST3070
     7              ISGS   ,NEXT   ,IXMP   ,ISGR   ,IWS    ,IES    ,IWC    ,     FAST3071
     8              IEC    ,IND    ,IU     ,IV     ,IW     ,ISI    ,IST    ,     FAST3072
     9              INRG   ,INSC   ,ISTP   ,INRP   ,INCP   ,IFXP   ,IFXS   ,     FAST3073
     1              IFXT   ,IFXE   ,IFXA                                         FAST3074
      NST = MMM                                                                  FAST3075
      STZ = VECTOR(YYY ,XXX,CP)                                                  FAST3076
      STS = STZ**2                                                               FAST3077
      PSI = -COSINE(CP,ZZZ)*STZ                                                  FAST3078
      CALL USTDUM(NEMOD,1,H(IRHO),H(ISSH),H(IMTL),H(ISGT),H(IND),                FAST3079
     1            H(IV),H(ISI),ST,NRG)                                           FAST3080
      TTT = STC                                                                  FAST3081
      MMM = NST                                                                  FAST3082
      USTAR = PDT                                                                FAST3083
      RETURN                                                                     FAST3084
      END                                                                        FAST3085
$IBFTC USTDM1  M94/2,XR7                                                         FAST3086
CUSTDUM$RANDOM SELECTION OF SCATTERING POINT FOR PROGRAM FASTER*T.JORDANFAST3087
      SUBROUTINE USTDUM(L1,L2,RHO,SSH,MTL,SGT,XM,TRA,TRB,ST,NRG)                 FAST3088
```

```
      DIMENSION ST(1),NRG(1),RHO(1),SSH(1),MTL(1),SGT(L1,L2),TRA(1),
     1          TRB(1),XM(1)
      COMMON/DUMUST/STZ    ,STS    ,PSI    ,STC    ,NST    ,POT
      COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BOC(3),ATA   ,ATB   ,ATC   ,
     1              ATD   ,AT    ,BT    ,AS    ,BS    ,JMIN  ,JMAX  ,
     2              KMIN  ,KMAX  ,JBAR  ,NZERO
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,
     2              NUNITX,NIMAX ,NMMAX ,NORDER,NDOWN ,INELAS,NTRANS,
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
     4              NOMAX ,NGMOD ,NOPENT,NOMOD ,NPOMAX,NPOMOD,NSCMAX,
     5              NVOMAX,NVOMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,
     6              NPRINT,NUNITS,NUMBER,KALIDE
      IF(MODELU.EC;0) GO TO 89
      STDB = SQRT(STS)
      XMTT = 0.0
      XMST = 0.0
      TRAB = 1.0
      STT = 0.0
      FMP = 0.0
      DOZ = 1.0/STS
      IF(NPOINT.GT.0) GO TO 5
      DOZ = 1.0
      IF(STDB.LE.DELTA) GC TC 5
      DOZ = (0.5*DELTA/STDB)**2
      DOZ = DOZ*(1.0 + DOZ)
    5 DOM = DOZ
      XMBB = 0.0
      DO 20 N=1,NST
      I = NRG(N)
      J = MTL(I)
      SGR = RHO(I)*SSH(JBAR)
      IF(J.GT.0) SGR = SGR + SGT(JBAR,J)
      STN = ST(N)
      STT = STT + STN
      STDS = STS + STT*(2.0*PSI + STT)
      STDA = STCB
      STDB = SQRT(STDS)
      XMST = XMST + AT*SGR*STN
      XMTT = XMTT + BT*SGR*(STDB - STDA)
      XMAA = XMBB
      XMBB = XMST + XMTT
      TRAA = TRAB
      DOM = 1.0/STCS
      IF(NPOINT.GT.0) GO TO 8
      DOM = 1.0
      IF(STDB.LE.DELTA) GO TO 8
```



```
      DOM = (0.5*DELTA/STCB)**2
      DOM = DOM*(1.0 + DOM)
    8 TRAB = DOM*EXP(-XMST - XMTT)/DOZ
      XMN = 0.0
      IF((SGR*STN).EQ.0.0) GO TO 10
      TRA(N) = SGR*TRAA
      IF(TRA(N).EC.0.0) GO TO 10
      ARGU = TRAB/TRAA
      IF(ARGU.GT.0.0) GO TO 9
      TRBN = (XMAA - XMBB)/STN
      XMN = TRA(N)*(EXP(XMAA - XMBB) - 1.0)/TRBN
      GO TO 11
    9 TRBN = ALOG(ARGU)/STN
      XMN = SGR*(TRAB - TRAA)/TRBN
   11 TRB(N) = TRBN
   10 XM(N) = XMN
   20 FMP = FMP + XMN
      IF(FMP.EQ.0.0) GO TO 95
   30 FMB = FMP*RANNO(NMB)
      STT = 0.0
      DO 40 N=1,NST
      IF(XM(N).GE.FMB) GO TO 50
      STT = STT + ST(N)
   40 FMB = FMB - XM(N)
   50 NST = N
      ARG = FMP
      TRAN = TRA(N)
      TRBN = TRB(N)
      VARG = FMB*TRBN + TRAN
      STP = ST(N)
      XYZ = VARG/TRAN
      IF(XYZ.GT.0.0) STP = ALOG(XYZ)/TRBN
      STC = STT + STP
      SGP = 1.0
      I=N
      GO TO 170
   89 CONTINUE
      PSIS=1.0
      NGT = 1
      TRA(1) = STZ
      STT = 0.0
      FBT = 0.0
   90 FMP = 0.0
      DO 120 N=1,NST
      I = NRG(N)
      J = MTL(I)
      SGR = RHO(I)*SSH(JBAR)
```


Astronuclear Laboratory

```
      IF(J.GT.0)SGR = SGR + SGT(JBAR,J)                                 FAST3183
      IF(NGT.GT.0)GO TO 100                                             FAST3184
      IF(SGR.EQ.0.0)GO TO 110                                           FAST3185
      XM(N) = ST(N)*SGR*ANG                                             FAST3186
      GO TO 120                                                         FAST3187
  100 STT = STT + ST(N)                                                 FAST3188
      TRA(N+1) = SQRT(STS + STT*(2.0*PSI + STT*PSIS))                   FAST3189
      TRB(N) = 0.0                                                      FAST3190
      IF(SGR.EQ.0.0)GO TO 110                                           FAST3191
      TRB(N) = BT*SGR*(TRA(N+1) - TRA(N))                               FAST3192
      FBT = FBT + TRB(N)                                                FAST3193
      XM(N) = TRB(N) + AT*SGR*ST(N)                                     FAST3194
      GO TO 120                                                         FAST3195
  110 XM(N) = 0.0                                                       FAST3196
  120 FMP = FMP + XM(N)                                                 FAST3197
      IF(FMP.GT.0.0)GO TO 125                                           FAST3198
   95 NST = 0                                                           FAST3199
      GO TO 195                                                         FAST3200
  125 AAA = EXP(-FMP)                                                   FAST3201
      ARG = 1.0 - AAA                                                   FAST3202
      VARG = 1.0 - ARG*RANNO(NMB)                                       FAST3203
      FMA = -ALOG(VARG)                                                 FAST3204
      FMB = FMA                                                         FAST3205
      FMC = 0.0                                                         FAST3206
      STC = 0.0                                                         FAST3207
      DO 140 I=1,NST                                                    FAST3208
      IF(FMB.LE.XM(I))GO TO 150                                         FAST3209
      FMB = FMB - XM(I)                                                 FAST3210
      STC = STC + ST(I)                                                 FAST3211
  140 FMC = FMC + TRB(I)                                                FAST3212
  150 NN = NRG(I)                                                       FAST3213
      MM = MTL(NN)                                                      FAST3214
      NST=I                                                             FAST3215
      SGP = RHO(NN)*SSH(JBAR)                                           FAST3216
      IF(MM.GT.0)SGP = SGP + SGT(JBAR,MM)                               FAST3217
      IF(NGT.GT.0)GO TO 160                                             FAST3218
      SGP = ANG*SGP                                                     FAST3219
      DDX = (FMP*AAA/ARG + FMA - 1.0)/ANG                               FAST3220
      STP = FMB/SGP                                                     FAST3221
      STC = STC + STP                                                   FAST3222
      GO TO 170                                                         FAST3223
  160 SGA = AT*SGP                                                      FAST3224
      SGB = BT*SGP                                                      FAST3225
      CQ = (FMB + SGA*STC + SGB*TRA(I))/SGB                             FAST3226
      AQ = SGA/SGB                                                      FAST3227
      FST = AQ**2 - PSIS                                                FAST3228
      BQ = (AQ*CQ + PSI)/FST                                            FAST3229
```



```
      HH =     (BQ**2 - (CQ**2 - STS)/FST)                              FAST3230
      IF(HH.LT.0.0) HH=0.0                                              FAST3231
      HH=SQRT(HH)                                                       FAST3232
      STQ = BQ - HH                                                     FAST3233
      IF(STQ.LT.STC)STQ = BQ + HH                                       FAST3234
      STP = STQ - STC                                                   FAST3235
      STC = STQ                                                         FAST3236
      ARE = CQ - AQ*STC                                                 FAST3237
      FMC = FMC + SGB*(ARE - TRA(I))                                    FAST3239
      PSI = (PSI + PSIS*STC)/ARE                                        FAST3240
      ANG =  AT + BT*PSI                                                FAST3241
      SGP = ANG*SGP                                                     FAST3242
  170 ST(I) = STP                                                       FAST3243
      PDT = ARG*STC**2/(SGP*VARG*12.566371)                             FAST3244
  195 RETURN                                                            FAST3245
      END                                                               FAST3246
$IBFTC VSTAR1  M94/2,XR7                                                FAST3247
CVSTAR *RANDOM SELECTION OF DIRECTION VECTOR FOR PROGRAM FASTER*T.JORDANFAST3248
      FUNCTION   VSTAR(CP,CC,C,AS,BS)                                   FAST3249
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX,   FAST3250
     1              NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,   FAST3251
     2              NUNITX,NIMAX ,NMMAX ,NCRDER,NDOWN ,INELAS,NTRANS,   FAST3252
     3              NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,   FAST3253
     4              NDMAX ,NGMCD ,MCMENT,NDMOD ,NPDMAX,NPDMOD,NSDMAX,   FAST3254
     5              NVDMAX,NVDMCD,NPCINT,MODELP,MODELQ,MODELU,MODELV,   FAST3255
     6              NPRINT,NUNITS,NUMBER,KALIDE                         FAST3256
      IF(MODELV - 1) 5,10,15                                            FAST3257
    5 CONTINUE                                                          FAST3258
      VSTAR = VSTDUM(CC,CP,C,BS,AS)                                     FAST3259
      GO TO 20                                                          FAST3260
   10 VSTAR = VSTDUM(CP,CC,C,AS,BS)                                     FAST3261
      GO TO 20                                                          FAST3262
   15 VSTAR = VSTDUM(CP,CC,C,AS + BS,0.0)                               FAST3263
   20 RETURN                                                            FAST3264
      END                                                               FAST3265
$IBFTC VSTDM1  M94/2,XR7                                                FAST3266
      FUNCTION VSTDUM(CP,CC,C,AS,BS)                                    FAST3267
      DATA PI/3.141593/                                                 FAST3268
      DIMENSION CP(3),CC(3),C(3),ROT(3,3),D(3)                          FAST3269
      EQUIVALENCE (D(1),D1),(D(2),D2),(D(3),D3)                         FAST3270
      CALL ROTATC(CP,ROT)                                               FAST3271
      DO 10 I=1,3                                                       FAST3272
      D(I) = 0.0                                                        FAST3273
      DO 10 J=1,3                                                       FAST3274
   10 D(I) = D(I) + ROT(I,J)*CC(J)                                      FAST3275
      THZ = 0.0                                                         FAST3276
      IF((ABS(D1) + ABS(D2)).GT.0.0) THZ = ATAN2(D2,D1)                 FAST3277
```

```
      SINZ = SQRT(1.0 - D3**2)
      PDA = SAMPLE(-1.0,1.0,-1.0,AS,D3)
      SINE = SQRT(1.0 - D3**2)
      PDA = PDA*SAMPLE(-PI,PI,0.,-BS*SINZ*SINE,THE)
      THE = THE + THZ
      D1 = SINE*COS(THE)
      D2 = SINE*SIN(THE)
      DO 20 I=1,3
      C(I) = 0.0
      DO 20 J=1,3
   20 C(I) = C(I) + D(J)*ROT(J,I)
      VSTDUM = PDA
      RETURN
      END
$IBFTC SAMPLR  M94/2,XR7
CSAMPLE*RANDOM SELECTION OF VARIABLE
      FUNCTION SAMPLE(VMN,VMX,VMD,ALP,VEE)
      IF(VMX.GT.VMN)GO TO 10
C DELTA FUNCTION (OR ERROR)
      PDF = 1.0
      VEE = VMN
      GO TO 30
   10 R = RANNO(NMB)
      IF(ALP.NE.0.0)GO TO 20
C UNIFORM DISTRIBUTION
      PDF = VMX - VMN
      VEE = VMN + R*PDF
      GO TO 30
C EXPONENTIAL DISTRIBUTION
   20 PLT = EXP(ALP*(VMD - VMN)) - 1.0
      PGT = EXP(ALP*(VMX - VMD)) - 1.0 + PLT
      PMD = PLT/PGT
      T = -1.0
      IF(R.GT.PMD)T = 1.0
      R = 1.0 + T*(R*PGT - PLT)
      VEE = VMD + T*ALOG(R)/ALP
      PDF = PGT/(R*ALP)
   30 SAMPLE = PDF
      RETURN
      END
$IBFTC RANDUM  M94/2,XR7
      FUNCTION RANNO(NMB)
      COMMON/CDCIBM/IBMCDC
      DATA N,NNN/0,11111/
      IF(IBMCDC.GT.0) GO TO 10
      R = RANF(NNN)
      GO TO 20
```



```
   10 CONTINUE
      IF(N.EQ.0) R = RANF(NNN)
      R = RANF(0)
   20 CONTINUE
      N = N + 1
      RANNO = R
      NMB = N
      RETURN
      END
$IBMAP RAN1
*       RANDOM NUMBER GENERATOR
        ENTRY    RANF
RANF    LDQ      L16
        MPY      L17
        LLS      4
        ALS      4
        LRS      4
        STQ      L16
        ADD      L16
        STO      L16
        ARS      4
        ORA      L20
        FAD      L20
        TRA      1,4
L16     OCT      002312421637
L17     OCT      000000001737
L20     OCT      200000000000
        END
$DATA
```


Astronuclear Laboratory