

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 THE
TRANSPORT EQUATION BY RANDOM SAMPLING

T. M. Jordar

DISTRIBUTED BY:



U.S. DEPARTMENT OF COMMERCE / NATIONAL BUREAU OF STANDARDS / INSTITUTE FOR APPLIED TECHNOLOGY



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

WANL-PR-(LL)-010

Westinghouse Astronuclear Laboratory



VOLUME 9

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

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)

FINAL PROGRESS REPORT

Volume 9

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

GPO PRICE \$ _____

CFSTI PRICE(S) \$ _____

Hard copy (HC) 4.3.00

Microfiche (MF) 1.65

11 653 July 65

ACTIVITY FORM 622	NAS-8-1350-4	
	(ACCESSION NUMBER)	(THRU)
	265	1
	(PAGES)	(COPIES)
	OR-91530	27
	(NASA, OR OR TRAC OR AD NUMBER)	(CATEGORY)

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.

ABSTRACT

This report is Volume 9 of nine volumes of the final report on "Synthesis of Computational 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

<u>Section</u>		<u>Page</u>
	ABSTRACT	iii
1.0	INTRODUCTION AND SUMMARY	1
2.0	THE MONTE CARLO METHOD	7
2.1	General Comments	7
2.2	The Order-of-Scatter Equations	9
2.3	Random Sampling Concept:	13
2.4	Inner Iterations	15
2.5	Sampling Functions	21
2.6	Point Angular Fluxes	27
2.7	Spatially Averaged Angular Fluxes	30
2.8	Concluding Remarks	34
3.0	PROGRAM LOGIC	37
3.1	Data Requirements	37
3.2	Program Flow	38
4.0	GEOMETRIC CONSIDERATIONS	43
4.1	Quadric Surfaces	43
4.2	Surface Calculations	43
4.3	Regions	47
4.4	Ray Tracing	49
5.0	FIXED SOURCES	53
5.1	Spatial and Angular Variables	53
5.2	Spatial and Angular Distributions	55

TABLE OF CONTENTS (CONTINUED)

<u>Section</u>	<u>Page</u>
5.3 Source Spectra	56
5.4 Fixed Source Acquisition	57
5.5 Source Evaluation	59
6.0 TRANSPORT AND SCATTERING KERNELS	63
6.1 Photon Cross Sections	63
6.2 Photon Transport	64
6.3 Photon Scattering	66
6.4 Neutron Cross Sections	67
6.5 Neutron Attenuation and Scattering	70
7.0 FLUX ESTIMATION	73
7.1 Angular Flux Contributions	73
7.2 Scalar Fluxes and Components	75
7.3 Flux Conversions	77
8.0 RANDOM SAMPLING TECHNIQUES	79
8.1 General Sampling Procedures	79
8.2 Spatial Sampling in the Fixed Source Coordinate System	82
8.3 Source Sampling Using a Pseudo Spherical Source	84
8.4 Separation of Variables for Scattering Point Selection	89
8.5 Direction to Scattering Point	93
8.6 Distance to Scattering Point	99
9.0 DATA INPUT INSTRUCTIONS	105
9.1 General Input Instructions	105
9.2 Section 1 Data, Title Cards, Limits, and Options	107

TABLE OF CONTENTS (CONTINUED)

<u>Section</u>	<u>Page</u>
9.3 Section 2 Data; Surfaces and Regions	115
9.4 Section 3 Data; Source Distributions	123
9.5 Section 4 Data; Microscopic Cross Sections	131
9.6 Section 5 Data; Detectors and Flux Conversions	138
9.7 Section 6 Data; Random Sampling Parameters	143
10.0 SAMPLE PROBLEM	154
11.0 REFERENCES	173
Appendix A Alternate Monte Carlo Integration Procedures	175
Appendix B Angular and Spatial Flux Moments	179
Appendix C Program Listing	183

PRECEDING PAGE BLANK NOT FILMED.



LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Schematic Diagram of the Final Design Method	2
2	The Monte Carlo Method - Inner Iterations	22
3	Problems in Region Descriptions	50
4	Direction Vector Rotations	61
5	Optimal Fixed Source Sampling	85
6	Optimal Scattered Source Sampling	90
7	Plane Surfaces	118
8	Conical Surfaces	119
9	Elliptical Cylinders and Ellipsoids	120
10	Source Distribution Variables	126
11	Sample Problem Geometry	152

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Special Surface Equations	117
2	Sample Problem Printout	153
3	Summary of Sample Problem Results	172

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

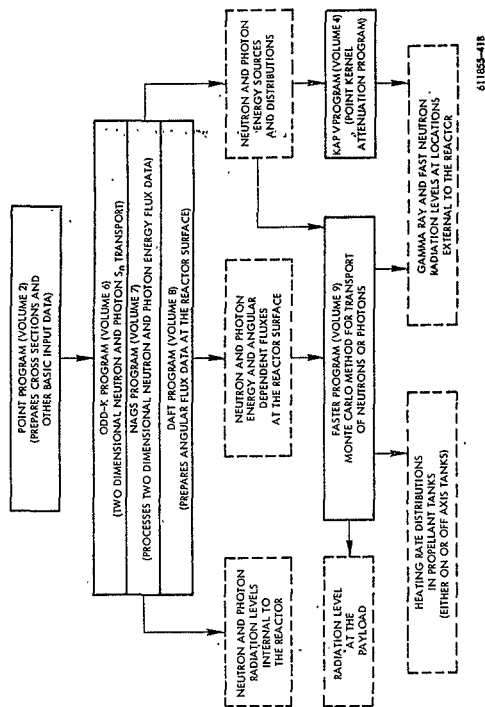


Figure 1. Schematic Diagram of the "Final" Design Method

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 quadratic 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 ($\pm$) 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 given in 10^{24} atoms/cm³ or gm/cm³,
- 2) microscopic cross sections can be in barns/atom or cm²/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:

$$\phi_0(\vec{r}, E) = \iiint S_0(\vec{r}', \vec{\Omega}', E) \frac{\exp \left[- \int_0^s \Sigma^t(\vec{r} - s' \vec{\Omega}', E) ds' \right]}{s^2} dV \quad (2.1)$$

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)

$S_0(\vec{r}, \vec{\Omega}, E)$ is the fixed differential source density

$$\left(\frac{\text{particles}}{\text{cm}^3 \text{ steradian Mev} \cdot \text{sec}} \right)$$

$\phi_0(\vec{r}, E)$ is the scalar flux $\left(\frac{\text{particles}}{\text{cm}^2 \text{ Mev} \cdot \text{sec}} \right)$

$\Sigma^t(\vec{r}, E)$ 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

$$\begin{aligned} \vec{r}' &= \vec{r} - s\hat{\Omega} \\ dV &= s^2 ds d\Omega \\ d\Omega &= d\mu d\theta \\ \mu &= \text{the cosine of the polar angle} \\ \theta &= \text{the azimuthal angle} \\ \phi_0(\vec{r}, E) &= \iint_{4\pi} \int_0^\infty \phi_0(\vec{r} - s\hat{\Omega}, \hat{\Omega}, E) \exp \left[-\int_0^s \Sigma'(\vec{r} - s'\hat{\Omega}, E) ds' \right] ds d\Omega \end{aligned} \quad (2.2)$$

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:

$$\phi_0(\vec{r}, \hat{\Omega}, E) = \int_0^\infty \phi_0(\vec{r} - s\hat{\Omega}, \hat{\Omega}, E) \exp \left[-\int_0^s \Sigma'(\vec{r} - s'\hat{\Omega}, E) ds' \right] ds \quad (2.3)$$

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

$$S_1(\vec{r}, \hat{\Omega}, E) = \iint_{4\pi} \int_0^\infty \phi_0(\vec{r}, \hat{\Omega}', E') \frac{d^2\Sigma}{d\Omega dE}(\vec{r}; \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) dE' d\Omega' \quad (2.4)$$

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

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:

$$\phi_k(\vec{r}, \hat{\Omega}, E) = \int_0^\infty S_k(\vec{r} - s\hat{\Omega}, \hat{\Omega}, E) \exp \left[-\int_0^s \Sigma'(\vec{r} - s'\hat{\Omega}, E) ds' \right] ds \quad (2.5)$$

$$S_{k+1}(\vec{r}, \hat{\Omega}, E) = \iint_{4\pi} \int_0^\infty \phi_k(\vec{r}, \hat{\Omega}', E') \frac{d^2\Sigma}{d\Omega dE}(\vec{r}; \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) dE' d\Omega' \quad (2.6)$$

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 $(k-1)$ th and lower order flux components, the unsolved portion of the problem, corresponding to the k th 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

$\phi_{\geq k}(\vec{r}, \hat{\Omega}, E)$ --the angular flux from particles having k or more collisions--and is given by a summation of equation 2.5:

$$\begin{aligned} \phi_{\geq k}(\vec{r}, \hat{\Omega}, E) &= \sum_{k'=k}^\infty \phi_{k'}(\vec{r}, \hat{\Omega}, E) \\ &= \int_0^\infty S_{\geq k}(\vec{r} - s\hat{\Omega}, \hat{\Omega}, E) \exp \left[-\int_0^s \Sigma'(\vec{r} - s'\hat{\Omega}, E) ds' \right] ds \end{aligned} \quad (2.7a)$$

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

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

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

$$\Phi_{\geq 0}(\vec{r}, \vec{\Omega}, E) = \int_0^S S_{\geq 0}(\vec{r}-s\vec{\Omega}, \vec{\Omega}, E) \exp \left[- \int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega}, E) ds' \right] ds \quad (2.7b)$$

where $\Phi_{\geq 0}(\vec{r}, \vec{\Omega}, E)$ is the total angular flux, $\phi(\vec{r}, \vec{\Omega}, E)$, from all orders-of-scatter and $S_{\geq 0}(\vec{r}, \vec{\Omega}, E)$ is the total differential source density, $S(\vec{r}, \vec{\Omega}, E)$, including the scattered source.

In computing the k th 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:

$$\phi_{\geq k}(\vec{r}, \vec{\Omega}, E) = \int_0^\infty \left[\int_{\vec{\Omega}'} \int_0^\infty S_k(\vec{r}-s\vec{\Omega}, \vec{\Omega}', E'; \vec{r}, \vec{\Omega}, E) d\vec{\Omega}' dE' \right] K(\vec{r}-s\vec{\Omega}, \vec{\Omega}, E) \exp \left[- \int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega}, E) ds' \right] ds \quad (2.8)$$

where the kernel $K(\vec{r}-s\vec{\Omega}, \vec{\Omega}, E; \vec{r}, \vec{\Omega}', E'; \vec{r}, \vec{\Omega}, E)$ 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:

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

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

$R(x)$ is the range of x

$$\left. \begin{aligned} p^*(x) &\geq 0 \\ p^*(x) &> 0 \text{ if } f(x) > 0 \\ \int_{R(x)} p^*(x) dx &= 1 \end{aligned} \right\} \quad (2.10)$$

The conditions imposed on $p^*(x)$ permit its use as a sampling function for obtaining n 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:

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

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

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

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

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

and the corresponding sample variance:

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

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

$$\begin{aligned}E[I_n] &= I \\ \sigma^2[I_n] &= \frac{1}{n} \sigma^2[f^*(x)] \\ E[V_n^2] &= \sigma^2[f^*(x)]\end{aligned}$$

The last two equations imply that:

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

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 k th 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 k th 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 $\vec{r}_{i,1}, \vec{r}_{i,2}, \dots, \vec{r}_{i,k}, \dots$, 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 $\vec{r}_{i,k}, k = 0, 1, \dots$, 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 k th order-of-scatter, will be retained.

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

$$\begin{aligned}\phi_k(\vec{r}, E) &= \iint_{4\pi} \phi_k(\vec{r}, \vec{\Omega}, E) d\Omega \\ &= \iint_{4\pi} \int_0^\infty S_k(\vec{r} - s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E) ds'\right] ds d\Omega \\ \phi_k(\vec{r}, E) &= \iiint S_k(\vec{r}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E) ds'\right] \frac{dV}{s^2} \quad (2.17)\end{aligned}$$

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:

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

with the restrictions:

$$\left. \begin{aligned} p_k^*(\vec{r}) &\geq 0 \\ p_k^*(\vec{r}) &> 0 \quad \text{if} \quad \iiint_{4\pi} \int_0^\infty S_k(\vec{r}, \vec{\Omega}, E) d\Omega ds > 0 \\ \iiint p_k^*(\vec{r}) dV &= 1 \end{aligned} \right\} \quad (2.19)$$

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 k th component of the scalar flux is given by:

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

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 k th scattered differential source density.

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

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 k th source distribution and can be evaluated for any desired direction $\vec{\Omega}$.

This procedure is also equivalent to representing the k th component of the differential source density by:

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

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 k th scattered source volume are required.

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

$$W_0^s(\vec{\Omega}, E) = \frac{S_0(\vec{r}_0, \vec{\Omega}, E)}{p_0^*(\vec{r}_0)} \quad (2.23)$$

The expected value of $S_0^*(\vec{r}, \vec{\Omega}, E)$ is just the fixed source distribution:

$$\begin{aligned} E \left[W_0^*(\vec{\Omega}, E, \delta(\vec{r} - \vec{r}_0)) \right] &= \iiint V_0 \left[W_0^*(\vec{\Omega}, E, \delta(\vec{r} - \vec{r}_0)) \right] P_0^*(\vec{r}_0) d\vec{r}_0 \\ &= \iiint V_0 \frac{S_0^*(\vec{r}_0, \vec{\Omega}, E) \delta(\vec{r} - \vec{r}_0) P_0^*(\vec{r}_0) d\vec{r}_0}{P_0^*(\vec{r}_0)} \\ &= S_0^*(\vec{r}, \vec{\Omega}, E) \end{aligned} \quad (2.24)$$

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 $W_{k-1}^*(\vec{\Omega}, E)$. Then the k th scattered energy angular point source at $\vec{r}_k$ is given by:

$$\begin{aligned} W_k^*(\vec{\Omega}, E) &= \frac{\iiint V_0 \phi_{k-1}(\vec{r}_k, \vec{\Omega}, E) \frac{d^2 \Sigma}{d\Omega dE}(\vec{r}_k, \vec{\Omega}, E \rightarrow \vec{\Omega}, E) dE' d\Omega'}{P_k^*(\vec{r}_k)} \\ &= \frac{\iiint V_0 \left[\int_0^{\infty} \frac{S_{k-1}^*(\vec{r}_k, \vec{\Omega}, E') \exp \left[-\int_0^s \Sigma^t(\vec{r}_k, \vec{\Omega}, E') ds' \right] \frac{d^2 \Sigma}{d\Omega dE}(\vec{r}_k, \vec{\Omega}, E' \rightarrow \vec{\Omega}, E) dE' d\Omega'}{P_k^*(\vec{r}_k)} \right]}{P_k^*(\vec{r}_k)} \\ &= \frac{\iiint V_0 \left[\frac{W_{k-1}^*(\vec{\Omega}, E') \exp \left[-\int_0^s \Sigma^t(\vec{r}_k, \vec{\Omega}, E') ds' \right] \frac{d^2 \Sigma}{d\Omega dE}(\vec{r}_k, \vec{\Omega}, E' \rightarrow \vec{\Omega}, E) dE'}{s^2 P_k^*(\vec{r}_k)} \right] \delta(\vec{r} - \vec{r}_{k-1}) dV}{s^2 P_k^*(\vec{r}_k)} \\ &= \frac{\int_0^{\infty} \left[\frac{W_{k-1}^*(\vec{\Omega}, E') \exp \left[-\int_0^s \Sigma^t(\vec{r}_k, \vec{\Omega}, E') ds' \right] \frac{d^2 \Sigma}{d\Omega dE}(\vec{r}_k, \vec{\Omega}, E' \rightarrow \vec{\Omega}, E) dE'}{s^2 P_k^*(\vec{r}_k)} \right]}{s^2 P_k^*(\vec{r}_k)} \end{aligned} \quad (2.25)$$

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 $W_{k-1}^*(\vec{\Omega}, E)$. 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:

$$W_{k-1}^\phi(E) = W_{k-1}^*(\vec{\Omega}_k, E) \frac{\exp \left[-\int_0^{s_k} \Sigma^t(\vec{r}_k, \vec{\Omega}_k, E) ds' \right]}{s_k^2 P_k^*(\vec{r}_k)} \quad (2.26)$$

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 k th scattered, angular point source:

$$W_k^*(\vec{\Omega}, E) = \int_0^{\infty} W_{k-1}^\phi(E') \frac{d^2 \Sigma}{d\Omega dE}(\vec{r}_k, \vec{\Omega}_k, E' \rightarrow \vec{\Omega}, E) dE' \quad (2.27)$$

The order of calculations may clarify this procedure:

- select $\vec{r}_k$ from $P_k^*(\vec{r})$
- calculate $s_k = |\vec{r}_k - \vec{r}_{k-1}|$
- calculate $\vec{\Omega}_k = (\vec{r}_k - \vec{r}_{k-1})/s_k$
- calculate $W_{k-1}^*(\vec{\Omega}_k, E)$ for the discrete direction $\vec{\Omega}_k$
 $k = 1$, $W_0^*(\vec{\Omega}_k, E)$ from equation 2.23 for $\vec{\Omega} = \vec{\Omega}_k$
 $k > 1$, $W_{k-1}^*(\vec{\Omega}_k, E)$ from equation 2.27 for $\vec{\Omega} = \vec{\Omega}_k$ using $W_{k-2}^\phi(E)$
- calculate $W_{k-1}^\phi(E)$ using equation 2.26.
- then $W_k^*(\vec{\Omega}, E)$ can be calculated, as required for a given direction $\vec{\Omega}$, by using equation 2.27.

Thus, $W_k^s(\vec{r}, \vec{\Omega}, E)$ 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 $W_{k-1}^s(E)$ 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:

$$\begin{aligned} E \left[W_1^s(\vec{r}, \vec{\Omega}, E) \delta(\vec{r} - \vec{r}_1) \right] &= \iiint_0 \iiint_1 \left[W_1^s(\vec{r}, \vec{\Omega}, E) \delta(\vec{r} - \vec{r}_1) \right] P_1^*(\vec{r}_1) P_0^*(\vec{r}_0) d\vec{r}_1 d\vec{r}_0 \\ &= \iiint_0 \iiint_1 \left\{ \int_0^\infty \frac{S_0(\vec{r}_0, \vec{\Omega}_1, E^1) \exp \left[-\int_0^1 (r_1 - s) \Sigma_1(\vec{r}_1, \vec{\Omega}_1, E^1) ds \right]}{s_1 P_0(\vec{r}_0) P_1(\vec{r}_1)} \right. \\ &\quad \times \left. \frac{d^2 \sum}{d\vec{\Omega} dE} (\vec{r}_1, \vec{\Omega}_1, E^1 \rightarrow \vec{\Omega}, E) dE^1 \right\} \delta(\vec{r} - \vec{r}_1) P_1^*(\vec{r}_1) P_0^*(\vec{r}_0) d\vec{r}_1 d\vec{r}_0 \\ &= \iiint_0 \left\{ \int_0^\infty S_0(\vec{r}_0, \vec{\Omega}, E) \frac{\exp \left[-\int_0^1 (\vec{r} - s) \Sigma_1(\vec{r}, \vec{\Omega}, E) ds \right]}{s_2} \frac{d^2 \sum}{d\vec{\Omega} dE} (\vec{r}, \vec{\Omega}, E \rightarrow \vec{\Omega}, E) dE \right\} d\vec{r}_0 \\ &= S_1(\vec{r}, \vec{\Omega}, E) \end{aligned} \quad (2.28)$$

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

$$\begin{aligned} E \left[W_k^s(\vec{r}, \vec{\Omega}, E) \delta(\vec{r} - \vec{r}_k) \right] &= \iiint_0 \dots \iiint_k W_k^s(\vec{r}, \vec{\Omega}, E) \delta(\vec{r} - \vec{r}_k) P_k^*(\vec{r}_k) \dots P_0^*(\vec{r}_0) d\vec{r}_k \dots d\vec{r}_0 \\ &= S_k(\vec{r}, \vec{\Omega}, E) \end{aligned}$$

and then showing that

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

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:

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

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

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

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

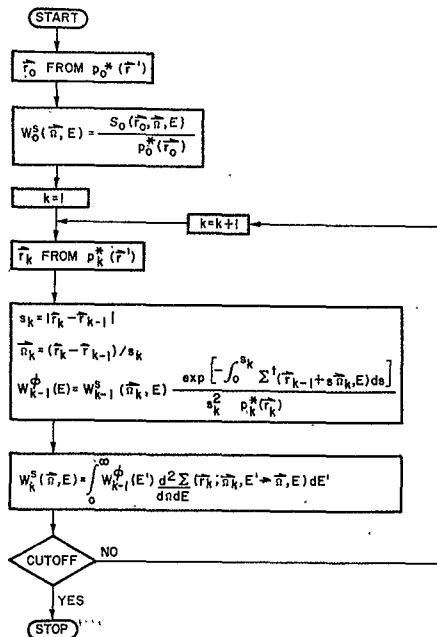


Figure 2. The Monte Carlo Method - Inner Iterations

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

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

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:

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

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

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:

$$\begin{aligned} D_{\geq k}(\vec{r}) &= \sum_{k'=k}^\infty \int_0^\infty f(E) \phi_{k'}(\vec{r}, E) dE \\ &= \sum_{k'=k}^\infty \int_0^\infty f(E) \iiint S_{k'}(\vec{r}', \vec{\Omega}, E) \exp \left[- \int_0^s \sum_i (\vec{r}' - s' \vec{\Omega}, E) ds' \right] dV dE \\ &= \iiint \left\{ \sum_{k'=k}^\infty S_{k'}(\vec{r}', \vec{\Omega}, E) \exp \left[- \int_0^s \sum_i (\vec{r}' - s' \vec{\Omega}, E) ds' \right] f(E) dE \right\} dV \\ &\approx \tilde{D}_{\geq k}(\vec{r}) \end{aligned} \quad (2.34)$$

where

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

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

$$\eta(E) = \int_0^1 \sum_i (\vec{r} - s' \vec{\Omega}', E) ds'$$

$K(\vec{\Omega}', E, \eta(E))$ 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:

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

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

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

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:

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

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

- it includes the spatial divergence ($1/s^2$) from the detector;
- it retains the exponential falloff of source point importance due to material attenuation, and
- 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:

$$p_k^*(\vec{r}') = \frac{1}{\tilde{D}_{\geq k}(\vec{r})} \int_0^\infty \int_{4\pi} \int_0^\infty \sum_{k=1}^s \tilde{D}_{\geq k-1}(\vec{\Omega}', E') \frac{d^2 \Sigma}{d\Omega dE}(\vec{r}' \rightarrow \vec{\Omega}', E' \rightarrow \vec{\Omega}, E) \quad (2.37a)$$

$$\frac{K(\vec{\Omega}', E, \eta(E)) \exp \left[\frac{-\int_0^1 (\vec{r}' - t' \vec{\Omega}', E) dt' - \eta(E)}{t'^2 s^2} \right] f(E) d\Omega' dE}{s^2}$$

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:

$$P_k^*(\vec{r}) = \frac{\exp \left[-\Sigma^t t - \Sigma^f s \right]}{t^2 s^2} \frac{1}{\iiint \exp \left[-\Sigma^t t - \Sigma^f s \right] dV} \quad (2.37b)$$

where Σ^t is the total cross section.

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

- it includes the spatial divergence from both the source ($1/t^2$) and the detector ($1/s^2$)
- it includes the exponential attenuation along both "legs" of the scattering triangle
- it includes the spatial and angular dependence of the scattering cross section, and
- 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:

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

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

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

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:

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

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.

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

This can be approximated using the above source density:

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

It is a straightforward task to show that the expected value of this approximation is, indeed, the total angular flux within an error $\bar{R}_0$ corresponding to the source errors $\bar{R}_0$ above:

$$E \left[\phi^* (\vec{r}, \vec{\Omega}, E) \right] = \phi (\vec{r}, \vec{\Omega}, E) - \bar{R}_0 \quad (2.43)$$

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:

$$\begin{aligned} \phi^* (\vec{r}, \vec{\Omega}, E) &= \int_0^\infty \frac{1}{n} \sum_{i=1}^n \left[\sum_{k=0}^\infty W_{ik}^s (\vec{\Omega}, E) \delta(\vec{r} - \vec{r}_{ik}) \right] \exp \left[- \int_0^s (\vec{r} - s' \vec{\Omega}, E) ds' \right] ds \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^\infty \int_0^\infty W_{ik}^s (\vec{\Omega}, E) \delta(\vec{r} - \vec{r}_{ik}) \exp \left[- \int_0^s (\vec{r} - s' \vec{\Omega}, E) ds' \right] ds \end{aligned} \quad (2.44)$$

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:

$$\Delta \phi_k^* (\vec{r}, \vec{\Omega}, E) = \int_0^\infty W_k^s (\vec{\Omega}, E) \delta(\vec{r} - \vec{r}_k) \exp \left[- \int_0^s (\vec{r} - s' \vec{\Omega}, E) ds' \right] ds \quad (2.45)$$

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:

$$\delta(\vec{r} - s\vec{\Omega} - \vec{r}_k) = \delta(s - s_k) \frac{\delta(\vec{\Omega} - \vec{\Omega}_k)}{s_k} \quad (2.46)$$

where:

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

Thus:

$$\begin{aligned} \Delta \phi_k^* (\vec{r}, \vec{\Omega}, E) &= \int_0^\infty W_k^s (\vec{\Omega}, E) \delta(s - s_k) \frac{\delta(\vec{\Omega} - \vec{\Omega}_k)}{s_k} \exp \left[- \int_0^s (\vec{r} - s' \vec{\Omega}, E) ds' \right] ds \\ &= W_k^s (\vec{\Omega}, E) \delta(\vec{\Omega} - \vec{\Omega}_k) \frac{\exp \left[- \int_0^{s_k} (\vec{r} - s' \vec{\Omega}, E) ds' \right]}{s_k^2} \end{aligned}$$

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:

$$\begin{aligned} \Delta \phi_k^* (\vec{r}, E) &= \int_4^\pi \int \Delta \phi_k^* (\vec{r}, \vec{\Omega}, E) d\Omega \\ &= W_k^s (\vec{\Omega}_k, E) \cdot \frac{\exp \left[- \int_0^{s_k} (\vec{r} - s' \vec{\Omega}_k, E) ds' \right]}{s_k^2} \end{aligned} \quad (2.48)$$

as one would expect.

As indicated in Section 2.4, the sampling functions which minimize the variance in the flux at the point $\vec{r}_k$ 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 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:

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

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 then transformed to an integration over solid angle about the point $\vec{r}_k$:

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

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:

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

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:

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

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

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

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

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:

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

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:

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

and if the volume has constant material properties:

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

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:

$$\begin{aligned} \Delta G_k^*(E) &= \iint_{4\pi} \Delta\phi_k^*(\vec{\Omega}, E) \left\{ \frac{A}{V} \right\} g(\vec{\Omega}) d\Omega \quad (2.57) \\ &= \iint_{4\pi} \left\{ \frac{\Delta\phi_k^*(\vec{\Omega}, E)}{q(\vec{\Omega})} \right\} \left\{ \frac{A}{V} \right\} g(\vec{\Omega}) d\Omega \\ &\approx \frac{1}{L} \sum_{\ell=1}^L \frac{\Delta\phi_k^*(\vec{\Omega}_\ell, E)}{q(\vec{\Omega}_\ell)} \left\{ \frac{A}{V} \right\} g(\vec{\Omega}_\ell) \end{aligned} \quad (2.58)$$

where $\left\{ \frac{A}{V} \right\}$ denotes either a surface or volume,

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

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:

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

$$\Delta\phi_k^*(\vec{\Omega}, E)_V = \frac{1}{V} \frac{1}{L} \sum_{j=1}^L \frac{W_k^j(\vec{\Omega}, E)}{q(\vec{\Omega}_j)} \int_{s \text{ in } V} \exp \left[- \int_0^s (\vec{r}_k + s' \vec{\Omega}, E) ds' \right] ds \delta(\vec{\Omega} - \vec{\Omega}_j) \quad (2.61)$$

for surfaces and volumes respectively since angular integrations with the arbitrary kernel $g(\vec{\Omega})$ 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-vold 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 $\pm$ 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 the 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:

$$u_i(\vec{r}) = a_{0,i} + a_{1,i}x + a_{2,i}y + a_{3,i}z + a_{4,i}x^2 + a_{5,i}y^2 + a_{6,i}z^2 + a_{7,i}xy + a_{8,i}yz + a_{9,i}zx \quad (4.1)$$

where $a_{j,i}$, $j = 0, 1, 2, \dots, 9$ 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,
 $\vec{\Omega} = (a, \beta, \gamma)$, a unit vector defining the direction of a straight line,
 or ray, emanating from $\vec{r}$, where a, 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
 $= (x + as, y + \beta s, z + \gamma s)$

Intersection of a Line and a Surface

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

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

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 :

$$v_i(\vec{r}, \vec{\Omega}) = \frac{1}{2} \begin{bmatrix} a a_{1,i} + \beta a_{2,i} + \gamma a_{3,i} \\ a x a_{4,i} + \beta y a_{5,i} + \gamma z a_{6,i} \\ (a y + \beta x) a_{7,i} + (\beta z + \gamma y) a_{8,i} + (\gamma x + a z) a_{9,i} \end{bmatrix} \quad (4.3)$$

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

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.

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

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

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

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

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

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:

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

Using the intersection equation 4.7 above:

$$\left. \frac{\partial u_i(\vec{r}')}{\partial s} \right|_{s=s_i} = 2 \left[v_i(\vec{r}, \vec{\Omega}) \pm \frac{-v_i(\vec{r}, \vec{\Omega}) \pm \sqrt{v_i^2(\vec{r}, \vec{\Omega}) - u_i(\vec{r}) w_i(\vec{r}, \vec{\Omega})}}{w_i(\vec{r}, \vec{\Omega})} \right] \\ = \pm 2 \sqrt{v_i^2(\vec{r}, \vec{\Omega}) - u_i(\vec{r}) w_i(\vec{r}, \vec{\Omega})} \quad (4.9)$$

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:

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

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 $\partial u_i(\vec{r}') / \partial s < 0$ 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:

$$\vec{n} = \frac{\nabla u_i(\vec{r}')}{|\nabla u_i(\vec{r}')|} = (c_1^n, c_2^n, c_3^n) \quad (4.11)$$

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

$$\begin{aligned} \vec{n} \cdot \Delta u_i(\vec{r}') &= [a_{1,i} + 2x' a_{4,i} + a_{7,i} y' + a_{9,i} z'] \vec{i} \\ &+ [a_{2,i} + 2y' a_{5,i} + a_{8,i} z' + a_{7,i} x'] \vec{j} \\ &+ [a_{3,i} + 2z' a_{6,i} + a_{9,i} x' + a_{8,i} y'] \vec{k} \\ &= c_1 \vec{i} + c_2 \vec{j} + c_3 \vec{k} \end{aligned} \quad (4.13)$$

$$c_i^n = \frac{C}{\left[\sum_{i=1}^3 c_i^2 \right]^{1/2}}, \quad i = 1, 2, 3 \quad (4.14)$$

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_i^H . 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:

$$k_{l,i} ; i = 1, 2, \dots \quad \text{where } k_{l,i} \text{ is the index of the surface forming the } i\text{th boundary of the region.}$$

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

$$\vec{r}_i^g = x_i^g \vec{e}_1 + y_i^g \vec{e}_2 + z_i^g \vec{e}_3 \quad \text{where } (x_i^g, y_i^g, z_i^g) \text{ are specified.}$$

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:

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

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 $\delta_{l,i} u_{k_{l,i}}(\vec{r}) < 0$ 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.

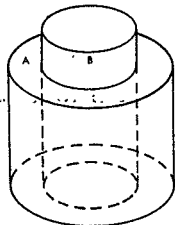
4.4 RAY TRACING

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.

$$s_{l+1} = \min_{i=1,2,\dots} \left\{ \begin{array}{l} s_{l,i} \text{ such that } s_{l,i} \geq s_1, \text{ and} \\ \text{if } s_{l,i} = s_1, \quad \delta_{l,i} \partial u_k(\vec{r} + s_1 \vec{n}) / \partial s > 0 \end{array} \right\} \quad (4.16)$$

s_1 = distance up to the region, and
 $s_{l,i}$ = is obtained from equation 4.6 or 4.7 for surface $k = k_{l,i}$
 using the sign of $\delta_{l,i}$ for non-planar surfaces.

3A. AMBIGUOUS REGION BOUNDARIES



3B. DISJOINT REGIONS

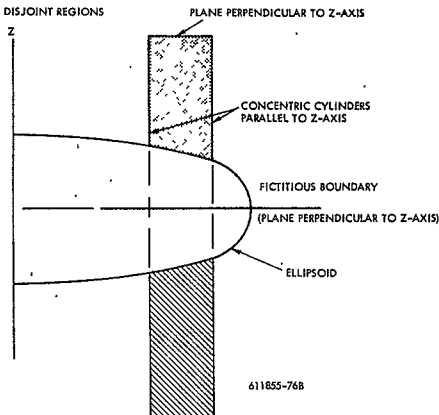


Figure 3. Problems in Region Descriptions

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:

- the next region have the surface just crossed as a boundary,
- the next region be on the other side of this surface (opposite signs on ambiguity indices), and
- for all other boundaries

$$u_{k'}(\vec{r} + s_{l+1}\vec{\Omega}) \cdot \vec{\Omega}_{l',i'} \leq 0, \quad k' = k_{l',i'}, \quad i' = 1, 2, \dots \quad (4.17)$$

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

- $n_k = 0$ if the k th surface has not been involved as yet, in the ray tracing,
- $n_k = 1$ if the constants $u_k(\vec{r})$, $v_k(\vec{r}, \vec{\Omega})$, $w_k(\vec{r}, \vec{\Omega})$ have been computed,
- $n_k = 2$ 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
- $n_k = 3$ 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:

$$\begin{aligned} v_1 &= r = \sqrt{x^2 + y^2}, \text{ the radius} \\ v_2 &= \theta = \tan^{-1}(y/x), \text{ the azimuthal angle measured} \\ &\quad \text{from the x-axis} \\ v_3 &= z, \text{ the axial coordinate} \end{aligned}$$

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

$$v_4 = \theta^1, \quad \text{the azimuthal angle measured from } \theta,$$

$$v_5 = \mu^1, \quad \text{the cosine of the polar angle measured from the z-axis.}$$

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:

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

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

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

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

$$v_4 = \theta^1, \quad \text{the azimuthal angle measured as shown in Figure 10}$$

$$v_5 = \mu^1, \quad \text{the cosine of the polar angle measured from the spherical radius vector.}$$

Source Translations

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

$$\vec{r} = (x + x_i, y + y_i, z + z_i)$$

5.2 SPATIAL AND ANGULAR DISTRIBUTIONS

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

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

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

Each variable v_i may take on only one value $v_{1,i} = v_{1,i}$ 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:

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

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:

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

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:

$$\text{Group } i : E_i \geq E \text{ (Mev)} \geq E_{i+1} \quad i = 1, 2, \dots \quad (5.3)$$

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:

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

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:

$$n_i^0 = \sum_{\substack{k=1 \\ E_{jk}^l < E_{jk}^h}} \int_{E_{jk}^l}^{E_{jk}^h} \eta_k(E) dE \quad (\text{particles in group } i) \quad (5.4)$$

$$\bar{E}_i^0 = \frac{1}{n_i^0} \sum_{\substack{k=1 \\ E_{jk}^l < E_{jk}^h}} \int_{E_{jk}^l}^{E_{jk}^h} \eta_k(E) E dE \quad (\text{average energy of the particles in group } i) \quad (5.5)$$

$$\text{where } \eta_k(E) = \frac{(E - E_{k+1}^l) \eta_k + (E_k - E) \eta_{k+1}}{E_k - E_{k+1}^l} \quad (5.6)$$

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

The groupwise number spectrum n_i^0 , $i = 1, 2, \dots$, 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 arrangement 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:

$$\begin{aligned}\phi^u(\vec{r}, \vec{\Omega}, E) &= \int_0^\infty S(\vec{r}-s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega}, E) ds'\right] ds \\ &= \int_0^{s_1} S(\vec{r}-s\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r}-s'\vec{\Omega}, E) ds'\right] ds \\ \phi^u(\vec{r}, \vec{\Omega}, E) &= \int_0^{s_1-s_0} S(\vec{r}-s_0\vec{\Omega}-t\vec{\Omega}, \vec{\Omega}, E) \exp\left[-\int_0^t \Sigma^t(\vec{r}-s_0\vec{\Omega}-t'\vec{\Omega}, E) dt'\right] dt \\ &= \phi(\vec{r}-s_0\vec{\Omega}, \vec{\Omega}, E)\end{aligned}\quad (5.8)$$

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:

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

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

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

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

e) calculate the angular variables v_4 and v_5

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

A linear interpolation of the source variable distributions yields:

$$W_i^s = \left[\frac{\sum_{j=1}^5 \frac{f(v_j)}{p_o(\vec{r}_o)} \right] n_i^o, \quad i = 1, 2, \dots \quad (5.10)$$

$$\vec{E}_i^s = \vec{E}_i^o$$

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

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

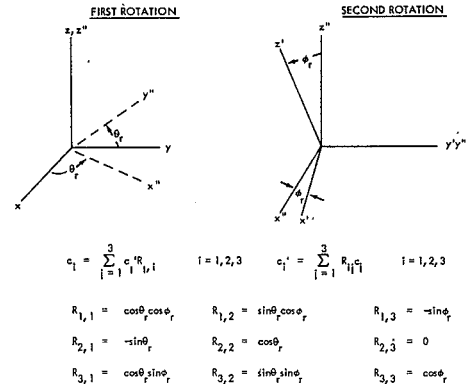
$$W_i^s = \int_{E_{i+1}}^{E_i} W_o^s(\vec{\Omega}, E) dE \quad (5.11)$$

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

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.

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



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

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

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

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

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	$\tan^{-1}\left(\frac{\sqrt{x^2 + y^2}}{z}\right)$

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 microscopic cross sections using input microscopic data. The two most common sets of units for microscopic data and compositions are both permitted in FASTEP:

$$\begin{aligned} u_x &= 0 \text{ for microscopic cross sections in barns/atom} \\ &= 1 \text{ for microscopic cross sections in cm}^2/\text{gm} \\ u_c &= 0 \text{ for compositions in } 10^{24} \text{ atoms/cm}^3 \\ &= 1 \text{ for compositions in gm/cm}^3 \end{aligned}$$

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 i th element is:

$$\begin{aligned} A_i &\text{ the atomic mass (a.m.u.) of the element} \\ Z_i &\text{ the atomic number of the element} \\ \rho_{m,i} &\text{ the density of the element in composite material } m \text{ with units according} \\ &\text{to } u_c \text{ and} \\ \sigma_{t,i}^j &\text{ the microscopic total cross section for energy level } j \text{ with units according} \\ &\text{to } u_x. \end{aligned}$$

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

$$\Sigma_{t,m}^j = \sum_{i=1}^I \left(\frac{0.6025}{A_i} \right)^{(u_c - u_x)} \rho_{m,i} \sigma_{t,i}^j \quad (\text{cm}^{-1}) \quad (6.1)$$

The total electron density is calculated by:

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

where $(0.6025/A_i)^u$ converts compositions to 10^{24} atoms/cm³, and $(0.6025/A_i)^u$ converts cross sections to barns/atom.

The energy absorption coefficient by energy level is computed by element and material. The absorption 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):

$$\mu_{i,i}^a = \sigma_{i,i}^t \left[\frac{1.995}{8} \left(\frac{0.6025}{A_i} \right)^u \left[\frac{\ln(1+2\eta)}{\eta^3} + \frac{2(1+\eta)(2\eta^2-2\eta-1)}{\eta^2(1+2\eta)} + \frac{8\eta^2}{3(1+2\eta)^3} \right] \right] \quad (6.3)$$

with units according to μ_x and where $\eta = E_x/0.511$.

6.2 PHOTON TRANSPORT

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

$$K_j^t(\vec{r}, \vec{r}') = \exp \left[- \int_0^s \Sigma^t(\vec{r} + s\vec{\Omega}; \vec{E}_j^s) ds \right], \quad s = |\vec{r} - \vec{r}'|, \quad \vec{\Omega} = (\vec{r} - \vec{r}')/s \quad (6.4)$$

This kernel is evaluated using $\vec{E}_j^s$, 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,

$$K_j^t(\vec{r}, \vec{r}') = \exp \left[- \sum_{i=1}^j \Sigma_i^v \Delta s_i \right] \quad (6.5)$$

where Δs_i is the distance in the i th 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

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

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:

$$\begin{aligned} W_i^\phi &= \int_{E_{i+1}}^{E_i} W_k^\phi(E) dE \\ \overline{E_i^\phi} &= \frac{1}{W_i^\phi} \int_{E_{i+1}}^{E_i} W_k^\phi(E) E dE \\ &\approx \overline{E_i^s} \end{aligned}$$

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 $\bar{E}_i^\phi$ so that the ratio of the energies before and after scattering for group i is:

$$R_i = \frac{0.511}{0.511 + \bar{E}_i^\phi (1 - \mu)} \quad (6.7)$$

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

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

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

$$\Delta W_i^s = W_i^\phi \frac{0.49875}{4\pi} N_e R_i^2 \left[R_i + \frac{1}{R_i} - 1 + \mu^2 \right] \quad (6.9)$$

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:

$$\left. \begin{aligned} W_{i'}^s &= \sum_{i \in J'} \Delta W_i^s \\ \bar{E}_{i'}^s &= \frac{1}{W_{i'}^s} \sum_{i \in J'} \Delta W_i^s R_i \bar{E}_i^\phi \end{aligned} \right\} i \in J' \text{ if } E_{i'} > R_i \bar{E}_i^\phi \geq E_{i'+1} \quad (6.10)$$

where $R_i \bar{E}_i^\phi$ is the average energy after scattering of particles originally in group i , and J' is the set of initial group indices i , for which this scattered energy is within the boundaries of group i' .

The angular point source at the scattering point is evaluated only for discrete directions $\hat{\Omega}$. 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 i th element is supplied to the FASTER program by energy group i as:

$$\begin{aligned} \sigma_{i,i}^t & \text{ the average total cross section for element } i \text{ and group } i \\ \sigma_{i \rightarrow k,i}^l & \text{ the } l\text{th Legendre expansion coefficient for element } i \text{ of the differential elastic scattering cross section for transfer from group } i \text{ to group } k, \\ \sigma_{i \rightarrow k,i}^{ne} & \text{ an isotropic, weighted, non-elastic transfer cross section} \\ \sigma_{i \rightarrow k,i}^{ne} & = \sigma_{i \rightarrow k,i}^{inelastic} + 2 \cdot \sigma_{i \rightarrow k,i}^{(n-2n)} + \dots \end{aligned}$$

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:

$$\sigma_{l,i}^0 = \sum_k \sigma_{l \rightarrow k,i}^0 \quad (6.12)$$

where $\sigma_{l,i}^0$ is the average total elastic scattering cross section for the l th group of the i th element

$$\sigma_{l,i}^1 = 3\bar{\mu}_{l,i} \sigma_{l,i}^0$$

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

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

$$\sigma_{l,i}^{tr} = \sigma_{l,i}^0 - \bar{\mu}_{l,i} \sigma_{l,i}^1 = \sigma_{l,i}^0 - \sigma_{l,i}^1/3 \quad (6.13)$$

The removal of this correction requires the equation:

$$\sigma_{l,i}^t = \sigma_{l,i}^{tr} + \sigma_{l,i}^1/3$$

If the total cross section is transport corrected, the corresponding correction to the group averaged elastic scattering cross section is applied to the l th-group term of the isotropic Legendre expansion coefficient, $\sigma_{l \rightarrow k,i}^0$ only:

$$\sigma_{l \rightarrow k,i}^{0, tr} = \sigma_{l \rightarrow k,i}^0 - \sigma_{l,i}^1/3 \quad (6.14)$$

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_i^h . 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 $\Sigma_{l,m}^t$ and the scattering cross sections $\Sigma_{l \rightarrow k,m}^s$ and $\Sigma_{l \rightarrow k,m}^{s,ne}$ 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:

$$\sigma_{l,i} \left[1 - \left(\frac{E_{out}}{E_{in}} \right) \right]_{l,i} = \frac{2A_i}{(A_i+1)^2} \sigma_{l,i}^0 \left(1 - \bar{\mu}_{l,i}^{c,m} \right) \quad (6.15)$$

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,

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

$\sigma_{l,i}^0$ is the total elastic scattering cross section as given by equation 6.12 for heavy elements ($A_i > 1$)
 $= \sigma_i^h$ for hydrogen

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

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

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

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

6.5 NEUTRON ATTENUATION AND SCATTERING

The neutron attenuation kernel $K_1^s(\vec{r}, \vec{r}')$ 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:

$$\Delta W_{1,i}^H = \sum_{j=1} W_j^\phi \left[\frac{1}{4\pi} \sum_{j'=j, m}^{ne} + \sum_{l=0} \sum_{j'=j, m}^l P_j(\mu) \right] \quad (6.17)$$

where m denotes the material at the scattering point

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

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

$$\left. \begin{aligned} P_0(\mu) &= 1 \\ P_1(\mu) &= \mu \\ P_l(\mu) &= \frac{1}{l!} \left[(2l-1) \mu P_{l-1}(\mu) - (l-1) P_{l-2}(\mu) \right] \end{aligned} \right\} \quad (6.18)$$

The neutron source for scattering from hydrogen is obtained only if $\mu > 0$, i.e., scattering in the laboratory coordinate system is restricted to angles < 90 degrees. Then the scattered source contribution is computed as:

$$\Delta W_{1,i}^h = 4\mu \rho^h \sum_{j=1} W_j^\phi \frac{\sigma_j^h}{4\pi} \quad (6.19)$$

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

$j \neq j'$ if $E_{1,i} > \mu^2 E_{1,i}^\phi \geq E_{1,i+1}$ i.e., if the scattered energy $\mu^2 E_{1,i}^\phi$ is within the boundaries of group j' ; 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:

$$W_{1,i}^s = \Delta W_{1,i}^H + \Delta W_{1,i}^h$$

$$\bar{E}_{1,i}^s = \frac{1}{W_{1,i}^s} \left[\bar{E}_{1,i}^n \Delta W_{1,i}^H + 4\mu \rho^h \sum_{j=1} W_j^\phi \frac{\sigma_j^h}{4\pi} \mu^2 \bar{E}_{1,i}^\phi \right] \quad (6.20)$$

where the $\bar{E}_{1,i}^n, i=1, 2, \dots$, 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 i th energy group from the k th inner iteration of the j th outer iteration is given by:

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

$$\Delta\phi_{ijk}^*(\vec{\Omega}) = \left\{ \frac{W_i s K^{\dagger}(\vec{r}_{ik}, \vec{r})}{|\vec{r} - \vec{r}_{ik}|^2} \right\} \delta(\vec{\Omega} - \vec{\Omega}_0) \quad (7.1)$$

$$\vec{\Omega}_0 = (\vec{r} - \vec{r}_{ik}) / |\vec{r} - \vec{r}_{ik}|$$

b) surface detector at a distance s from $\vec{r}_{ik}$ along $\vec{\Omega}_1$

$$\Delta\phi_{ijk}^*(\vec{\Omega}) = \left\{ \frac{W_i s K^{\dagger}(\vec{r}_{ik}, \vec{r}_{ik} + s\vec{\Omega}_1)}{A \cdot L \cdot |\vec{\Omega}_1 \cdot \vec{n}| \cdot q^*(\vec{\Omega}_1)} \right\} \delta(\vec{\Omega} - \vec{\Omega}_0) \quad (7.2)$$

$$\vec{\Omega}_0 = \vec{\Omega}_1$$

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

$$\Delta\phi_{ijk}^*(\vec{\Omega}) = \left\{ \frac{W_i^s K_i^{-1} (\vec{r}_{ik} \cdot \vec{r}_{ik} + s\Omega_i) \Delta s}{V \cdot L \cdot q(\vec{\Omega}_i)} \right\} \delta(\vec{\Omega} - \vec{\Omega}_0) \quad (7.3)$$

$$\vec{\Omega}_0 = \vec{\Omega}_i$$

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

$$\Delta\phi_{ijk}^*(\vec{\Omega}) = \left\{ \frac{W_i^s K_i^{-1} (\vec{r}_{ik} \cdot \vec{r}_{ik} + s\vec{\Omega}_i)}{V \cdot L \cdot q(\vec{\Omega}_i)} \cdot \frac{1 - \exp[-\Delta s \Sigma_i^v]}{\Sigma_i^v} \right\} \delta(\vec{\Omega} - \vec{\Omega}_0) \quad (7.4)$$

$$\vec{\Omega}_0 = \vec{\Omega}_i$$

or in general:

$$\Delta\phi_{ijk}^*(\vec{\Omega}) = \Delta\phi_{ijk} \delta(\vec{\Omega} - \vec{\Omega}_0) \quad (7.5)$$

where $\Delta\phi_{ijk}$ is the quantity in brackets $\left\{ \right\}$ above. The angular dependence implied by the Dirac delta function is only a formality 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 $\vec{\Omega}_0$. Surface and volume detectors may receive more than one flux contribution. This may occur if more than one discrete direction, $\vec{\Omega}_i$, ($L > 1$) 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:

$$\phi_j^0 = \frac{1}{n} \sum_{i=1}^n \sum_{k=0} \iint_{4\pi} \Delta\phi_{ijk} \delta(\vec{\Omega} - \vec{\Omega}_0) d\Omega \quad (7.6)$$

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

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

$$v_j^2 = \frac{1}{n-1} \left[\sum_{i=1}^n \left(\sum_{k=0} \Delta\phi_{ijk} \right)^2 - n (\phi_j^0)^2 \right] \quad (7.7)$$

Relative Error

$$E_j = \frac{\sqrt{v_j^2/n}}{\phi_j^0} \quad (7.8)$$

Differential Number Flux

$$\phi_i^0 / (E_i - E_{i+1}) \quad (7.9)$$

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

$$\sum_{i=1}^I \phi_i^0 \quad (7.10)$$

Energy Flux

$$I_i^0 = \frac{1}{n} \sum_{i=1}^n \overline{E_{ijk}^s} \Delta \phi_{ijk} \quad (7.11)$$

Average Group Energy

$$\bar{E}_i = I_i^0 / \phi_i^0 \quad (7.12)$$

Differential Energy Flux

$$I_i^0 / (E_i - E_{i+1}) \quad (7.13)$$

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

$$\sum_{i=1}^I I_i^0 \quad (7.14)$$

Average Flux Between Outer Iterations n_1 and n_2

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

Flux Contribution from m th Fixed Source

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

Flux Contribution from l th Scattering Region

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

Flux Contribution from k th Order of Scatter

$$\frac{1}{n} \sum_{i=1}^n \Delta \phi_{ijk} \quad (7)$$

Azimuthally averaged Legendre moments of the angular flux can also be obtained 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 equations for externally manipulating these moments are summarized in Appendix B.

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

7.3 FLUX CONVERSIONS

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

$$D_i = \phi_i^0 \frac{f_i (\bar{E}_i - E_{i+1}) + f_{i+1} (E_i - \bar{E}_i)}{E_i - E_{i+1}} \quad (7)$$

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

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

$$D = \sum_{i=1}^n D_i \quad (7.20)$$

$$E_{\min} = \frac{1}{D} \left[\sum_{i=1}^n E_i^2 D_i^2 \right]^{1/2} \quad (7.21)$$

$$E_{\max} = \frac{1}{D} \sum_{i=1}^n E_i D_i \quad (7.22)$$

The equation for $E_{\max}$ is obtained by applying the Schwarz inequality to the covariance 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:

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

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

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

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:

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

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 then written, for x on the interval $(a, b]$ as $p^*(x; a, b, a, x_0)$ with the following special cases:

$$\begin{aligned} \text{a) } p^*(x) &= \delta(x - a) \text{ if } a = b \\ \text{b) } p^*(x) &= \frac{1}{b-a} \text{ if } a < b \text{ and } a = 0 \\ \text{c) } p^*(x) &= \frac{a}{c} \exp[a|x - x_0|] \text{ if } a < b \text{ and } a \neq 0 \end{aligned} \quad (8.4)$$

where, for $a \neq 0$

x_0 is the preferred value of x ,

$$a \leq x_0 \leq b$$

$$C = A + B$$

$$A = a \int_a^{x_0} \exp[a|x - x_0|] dx = \exp[a|x - x_0|] - 1$$

$$B = a \int_{x_0}^b \exp[a|x - x_0|] dx = \exp[a|x - x_0|] - 1 \quad (8.5)$$

The parameter a is usually defined from a specification of the relative importance, ρ , of x_0 as compared to the point, a or b , farthest away from x_0 .

$$\rho = 1/\exp[a(x_0 - a)] \text{ or } \rho = 1/\exp[a(b - x_0)], \text{ i.e.,}$$

$$a = - \frac{\ln \rho}{\max(x_0 - a, b - x_0)} \quad (8.6)$$

The point x_0 is "least preferred" if ρ is less than unity.

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

$$\begin{aligned} \text{a) if } a &= b \\ \text{set } x &= a \\ \text{set } p^*(x) &= 1.0 \quad [\text{actually equal to } \delta(x - a)] \end{aligned} \quad (8.7)$$

$$\begin{aligned} \text{b) if } a < b \text{ and } a &= 0 \\ \text{obtain } \xi &\text{ from RANNO} \\ \text{solve equation 8.3,} \\ \xi &= \int_a^x \frac{dx'}{b-a} = \frac{x-a}{b-a} \\ \text{i.e., } x &= a + \xi(b-a) \\ p^*(x) &= 1/(b-a) \end{aligned} \quad (8.8)$$

$$\begin{aligned} \text{c) if } a < b \text{ and } a &\neq 0 \\ \text{obtain } \xi &\text{ from RANNO} \\ \text{solve equation 8.3,} \\ \text{i.e., } \xi &= \frac{a}{c} \int_a^x \exp[a|x - x_0|] dx \\ 1) \text{ if } \xi < A/C, \text{ then } x &\leq x_0 \\ \text{and } \xi &= \frac{1}{C} \left\{ A - a \int_x^{x_0} \exp[a|x' - x_0|] dx' \right\} \\ \text{or } \xi C - A &= - \left\{ \exp[-a(x - x_0)] - 1 \right\} \\ 2) \text{ if } \xi > A/C, \text{ then } x &> x_0 \\ \text{and } \xi &= \frac{1}{C} \left\{ A + a \int_{x_0}^x \exp[a|x' - x_0|] dx' \right\} \\ \text{or } \xi C - A &= \left\{ \exp[a(x - x_0)] - 1 \right\} \end{aligned}$$

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

$$\text{then } \exp[\delta \alpha(x - x_o)] = 1 + \delta(\xi C - A)$$

$$\text{[e]} \quad p = \frac{1}{\alpha} \delta(\xi C - A) \cdot \dots$$

$$\text{then } x = x_o + \frac{\delta \ln D}{\alpha}$$

$$\text{and } p^*(x) = \frac{\alpha D}{C}$$

(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:

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

where p_i^* is the input relative importance of the i th source

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

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

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

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

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

$$\sum_{i=1}^{i-1} p_i^* < \xi \leq \sum_{i=1}^i p_i^* \quad (8.14)$$

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

$$q_i^*(\vec{r}) dV = \frac{3}{i-1} p_i^* \left(v_i; v_i^{\min}, v_i^{\max}, v_i^o, a_i \right) \frac{dV}{v_i^n} \quad (8.15)$$

where $n = 1$ and 2 for cylindrical and spherical geometries, respectively
 $= 0$ for rectangular geometries

$v_i^{\min}$ is the minimum value of the i th variable

$v_i^{\max}$ is the maximum value of the i th variable

v_i^o is the preferred value of the i th variable

$$a_i = \frac{-\ln p_i^*}{\max(v_i^o - v_i^{\min}, v_i^{\max} - v_i^o)}$$

p_i^* is the input relative importance of v_i^o

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

$$\frac{dv}{v_1^n} = \left(v_1^n dv_1 dv_2 dv_3 \right) / v_1^n = dv_1 dv_2 dv_3 \quad (8.16)$$

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 i th volume, the value of $p_o^*(\vec{r}_o)$ is:

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

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.

$$p_o^*(\vec{r}) dV = \frac{u^*(\vec{\Omega}) v^*(\vec{r}; \vec{\Omega})}{s^2} (s^2 ds d\Omega) \quad (8.18)$$

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 $\vec{\Omega}^1$ in this rotated coordinate system uses the sampling function:

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

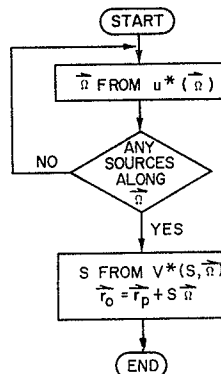
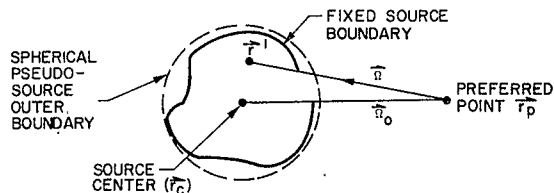


Figure 5. Optimal Fixed Source Sampling

i.e., discrete values of μ' and θ' 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:

$$\mu^{\min} = -1 \text{ if } R > R_0 = \left| \vec{r}_p - \vec{r}_c \right|$$

$$= \left[R_0^2 - R^2 \right]^{1/2} \text{ if } R \leq R_0 \quad (8.20)$$

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

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

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

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

$$\vec{\Omega}' = c_1 \vec{i}' + c_2 \vec{j}' + c_3 \vec{k}', \quad c_1' = \sqrt{1 - \mu'^2} \cos \theta'$$

$$c_2' = \sqrt{1 - \mu'^2} \sin \theta'$$

$$c_3' = \mu' \quad (8.22)$$

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

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

The rotation matrix R_{ji} is obtained as shown in Figure 4 for the rotation angles:

$$\phi_r = \cos^{-1} (c_3^0)$$

$$\theta_r = \tan^{-1} (c_2^0 / c_1^0) \quad (8.24)$$

where $\vec{c}_0 = c_1^0 \vec{i} + c_2^0 \vec{j} + c_3^0 \vec{k} = (\vec{r}_c - \vec{r}_p) / k$, the unit vector directed towards the center of the sphere enclosing the fixed sources.

Spatial Dependence

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 $s \geq 0$. 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 random 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 i th region along $\vec{\Omega}$ as:

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

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

$$A_i = I_i p_{i1} (\vec{r}_p + (s_i + \epsilon) \vec{\Omega}) \exp \left[-\int_0^{s_i} \sum_{j=1}^J (\vec{r}_p + s' \vec{\Omega}) ds' \right] \quad (8.26)$$

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

$$\alpha_i = \beta_i \sum_{j=1}^J (\vec{r}_p + \vec{s}_j \vec{\Omega}) + \frac{2}{s_{i+1} - s_i} \ln \left[\frac{p_{i1} (\vec{r}_p + (s_i + \epsilon) \vec{\Omega})}{p_{i1} (\vec{r}_p + s_i \vec{\Omega})} \right] \quad (8.27)$$

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

$p_i, (\vec{r}_p + (s_i + \epsilon)\vec{\Omega})$ is the normalized spatial source density just inside the region,
i.e., $\epsilon = \frac{s_{i+1} - s_i}{100}$

$p_i, (\vec{r}_p + \vec{s}_i\vec{\Omega})$ is the normalized spatial source density half way through the region

I_{i_1} is the total source intensity (Mev/sec),

$\hat{\Omega}_0$ 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, $\hat{\Omega}_0$, corresponds to an average source energy $\bar{E}_p$ computed for the preferred point $\vec{r}_p$ by subroutine GROUP as:

$$E_{\hat{\Omega}_0} > \bar{E}_p \geq E_{\hat{\Omega}_0 - 1}$$

$$\text{where } \bar{E}_p = \frac{\sum_i G_i \bar{E}_i}{\sum_i G_i}, \quad G_i = \frac{g_i n_i^0 \exp[-\eta_i] (1 + b_i \eta_i)}{\sum_i \eta_i} \quad (8.28)$$

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

b_i is an input linear buildup coefficient,

$n_i^0, \bar{E}_i^0$ 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_V is the cross section for the region over which the source is superimposed (≈ 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.

$$\xi = \int_0^s v^*(s'; \vec{\Omega}) ds' \\ = \frac{\sum_{i=1}^{k-1} \frac{A_i}{a_i} [1 - \exp[-a_i(s_i + 1 - s_i)]]}{\sum_{i=1}^k \frac{A_i}{a_i} [1 - \exp[-a_i(s_{i+1} - s_i)]]} + \int_{s_k}^s A_k \exp[-a_k(s' - s_k)] ds' \quad (8.29)$$

$$s = s_k - \frac{1}{a_k} \ln \left[1 - \frac{a_k}{A_k} (\xi A_{>0} - A_{<k}) \right] \\ v^*(s; \vec{\Omega}) = \left\{ \begin{aligned} &A_k - a_k (\xi A_{>0} - A_{<k}) / A_{>0} \\ &A_{<k} \end{aligned} \right\} \quad (8.30)$$

where $A_{>0} = \sum_{i=1}^k \frac{A_i}{a_i} [1 - \exp[-a_i(s_{i+1} - s_i)]]$, the denominator of equation 8.29

$$A_{<k} = \sum_{i=1}^{k-1} \frac{A_i}{a_i} [1 - \exp[-a_i(s_{i+1} - s_i)]]$$

that part of the numerator of equation 8.29 up to the region containing the source point

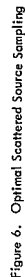
Thus the discrete position reactor is given by-

$$\vec{r}_0 = \vec{r}_p + s \vec{\Omega}$$

$$\text{with } p_0^*(\vec{r}_0) = \frac{u^*(\vec{\Omega}) v^*(s; \vec{\Omega})}{2} \quad (8.31)$$

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 k th 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 $\vec{r}$.



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:

where both $q^*(\vec{\Omega}_i)$ and $u^*(x)$ 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 $\vec{r}_{k-1}$:

$$E_i^{\wedge} > \hat{E} \geq E_{i+1}^{\wedge}$$

$$\bar{E} = \frac{\sum_{\text{detectors}} \sum_i \frac{\Delta\phi_{i,j,k-1}}{\Delta\phi_{i,j,k-1}} g_{i,j,b} \bar{E}_i^3}{\sum_{\text{detectors}} \sum_i \frac{\Delta\phi_{i,j,k-1}}{\Delta\phi_{i,j,k-1}} g_{i,j,b}} \quad (8.33)$$

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 $\hat{\Omega}_i$ are randomly selected from the angular sampling function $q^*(\hat{\Omega})$.

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. $dV = r^2 d\Omega$. 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, $1/r^2$ 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:

$$u^*(\vec{r}; \Omega) dV = \left\{ p_1^* u_1^*(\vec{r}) + p_2^* u_2^*(\vec{r}) \right\} dV \quad (8.34)$$

where

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

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

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:

$$\begin{aligned} p_1^* &= \eta_1 / \eta, \text{ the assumed importance of volume } V_1 \\ p_2^* &= \eta_2 / \eta, \text{ the assumed importance of volume } V_2 \\ \eta &= \eta_1 + \eta_2 \end{aligned}$$

$$\begin{aligned} \text{where } \eta_1 &= \int_0^{h/2} \sum_{k=1}^i (\vec{r}_{k-1} + s' \vec{\Omega}_k) ds', \text{ an average number of mean free paths in volume } V_1 \\ \eta_2 &= \int_0^{h/2} \sum_{k=1}^i (\vec{r}_p - s' \vec{\Omega}_k) ds', \text{ an average number of mean free paths in volume } V_2 \end{aligned} \quad (8.35)$$

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

$$V_1 \text{ if } \xi = \iiint_{V_k} (\vec{r}') dV = f \cdot p_1^* \text{ i.e. } \xi < p_1^*$$

$$V_2 \text{ if } \xi = p_1^* + f' p_2^* \text{ i.e. } \xi > p_1^* \quad (8.36)$$

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 sub-routine QSTAR using SAMPLE

$$q^*(\hat{n}) d\Omega^* = \sum_{i=4}^5 p^*(v_i; v_i^{\min}, v_i^{\max}, v_i^0, a_i) dv_i \quad (8.37)$$

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.

Direction Vector Selection - General

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:

$$f_a(\hat{n} \cdot \hat{n}_{k-1}) = f_a(\mu) = \exp [a_s \bar{a}_s \mu]$$

$$\left. \begin{aligned} \bar{a}_s &= \left(\rho_A^h a_A^h - \sum_A^H a_A^H \right) / \left(\rho_A^h a_A^h + \sum_A^H a_A^H \right) \text{ in subroutine SOBER} \\ \bar{a}_s &= \left(\eta^h a_A^h + (\eta - \eta^h) a_A^H \right) / \eta \text{ in subroutine SOLVER} \end{aligned} \right\} \quad (8.38)$$

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

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

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

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

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

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

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

a_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.

$$f_p(\hat{n} \cdot \hat{n}_p) = \exp(a_p \bar{a}_p \hat{n} \cdot \hat{n}_p) \quad (8.39)$$

where:

a_p is an input adjustment factor,

$\bar{a}_p$ is an internally computed number

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

$$f_p(\mu) = \int_0^{\infty} \max \frac{\exp[-\bar{S}(s+\mu)]}{s^2} s^2 ds \quad (8.40)$$

where

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

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

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

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

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

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

where Schwarz' inequality was used.

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

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

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

$$\bar{a}_p = \frac{\eta}{2} - \frac{1}{3} \quad (8.42)$$

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

$$q^*(\hat{n}) = p^*(\mu'; -1, 1, -1, \alpha_p \bar{a}_p) \\ p^*(\theta'_0; -\pi, \pi, 0, -\alpha_s \bar{a}_s, \sqrt{1-\mu'^2} \sqrt{1-\hat{n}_{k-1} \cdot \hat{n}_p^2}) \quad (8.43)$$

where

$$\hat{n}_r = \hat{n}_p$$

$$\mu' = \hat{n}^i \cdot \hat{n}_r$$

and θ'_0 is the azimuthal angle between the $\hat{i}^i$ axis and the projection of $\hat{n}_{k-1}$ in the $x'-y'$ plane and where the rotated coordinate system has $\hat{k}' = \hat{n}_p$.

b) Original Direction Most Important

$$q^*(\hat{n}) = p^*(\mu'; -1, 1, -1, \alpha_s \bar{a}_s) \\ p^*(\theta'_0; -\pi, \pi, 0, -\alpha_p \bar{a}_p, \sqrt{1-\mu'^2} \sqrt{1-\hat{n}_{k-1} \cdot \hat{n}_p^2}) \quad (8.44)$$

where

$$\hat{n}_r = \hat{n}_{k-1}$$

$$\mu' = \hat{n}^i \cdot \hat{n}_r$$

θ'_0 is the azimuthal angle between the $\hat{i}^i$ axis and the projection of $\hat{n}_p$ in the $x'-y'$ plane, and where the rotated coordinate system has $\hat{k}' = \hat{n}_{k-1}$.

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:

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

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

Assuming an exponential variation $\exp [\bar{a}_s \mu]$ for intermediate angles μ implies $\bar{a}_s = \bar{a}_s$ (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

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

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 techniques.

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 $\vec{r}_b = |\vec{r}_b - \vec{r}_a|$

$\vec{\Omega}$ 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.

$\Sigma_{\uparrow}^V(s)$ is the total cross section at a distance s from $\vec{r}_0$ for the energy group $\uparrow$, also denoted by $\Sigma_{\uparrow, i}^V$, 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 $\vec{r}_0$).

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

Exponential Transformation

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:

$$u^*(s) = \frac{-\frac{d}{ds} \exp \left[-\int_0^s \Sigma_{\uparrow}^V(s') [a - \beta \mu(s')] ds' \right]}{1 - \exp \left[-\int_0^s \Sigma_{\uparrow}^V(s') [a - \beta \mu(s')] ds' \right]} \quad (8.47)$$

where

$\mu(s)$ is the cosine of the scattering angle to $\vec{r}_b$ at a distance s from

$$\vec{r}_0, \text{ i.e., at } \vec{r} = \vec{r}_0 + s\hat{\Omega}$$

$$\mu(s) = \frac{\mu_0 h - s}{t}$$

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

$$= \sqrt{h^2 + s^2 - 2\mu_0 sh}$$

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

$$u^*(s) = \frac{1}{C} \Sigma_{\uparrow}^V(s) [a - \beta \mu(s)] \exp \left[-\int_0^s \Sigma_{\uparrow}^V(s') [a - \beta \mu(s')] ds' \right] \quad (8.48)$$

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

$$\begin{aligned} \Sigma_{\uparrow}^V(s) [a - \beta \mu(s)] &= \Sigma_{\uparrow}^V(s) \left(a - \beta \frac{\mu_0 h - s}{t} \right) \\ &= \Sigma_{\uparrow}^V(s) \frac{[a t + \beta s] - \beta \mu_0 h}{t} \end{aligned} \quad (8.49)$$

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:

$$\begin{aligned} \int_0^s \Sigma_{\uparrow}^V(s') [a - \beta \mu(s')] ds &= a \sum_i \Delta s_i \Sigma_{\uparrow, i}^V - \beta \sum_i \Sigma_{\uparrow, i}^V \int_{s_i}^{s_{i+1}} \mu(s) ds \\ &= a \sum_i \Delta s_i \Sigma_{\uparrow, i}^V + \beta \sum_i \Delta s_i \Sigma_{\uparrow, i}^V \end{aligned} \quad (8.50)$$

where

$$\begin{aligned} \Delta s_i &\text{ is the incremental distance in the } i\text{th region along } \hat{\Omega} \\ &= s_{i+1} - s_i \end{aligned}$$

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:

$$\xi = \frac{1}{C} \left[1 - \exp \left(- \int_0^s \Sigma_{\uparrow}^{\vee}(s') (a - \beta \mu(s')) ds' \right) \right]$$

$$\text{i.e. } \exp \left[- \int_0^s \Sigma_{\uparrow}^{\vee}(s') (a - \beta \mu(s')) ds' \right] = 1 - \xi C$$

$$\text{or } [a(s - s_i) + \beta(t - t_i)] \Sigma_{\uparrow, i}^{\vee} + \sum_{i=1}^{i-1} \Sigma_{\uparrow, i}^{\vee} (\alpha \Delta s_i + \beta \Delta t_i) = -\ln(1 - \xi C)$$

This equation can be written in the form

$$t = As + B$$

where

$$A = -\frac{\alpha}{\beta}$$

$$B = -\frac{1}{\beta} \left\{ \alpha s_i + \beta t_i - \frac{1}{\Sigma_{\uparrow, i}^{\vee}} \left[\ln(1 - \xi C) + \sum_{j=1}^{i-1} \Sigma_{\uparrow, j}^{\vee} (\alpha \Delta s_j + \beta \Delta t_j) \right] \right\}$$

$$\text{But } t^2 = s^2 + h^2 - 2\mu_o sh = A^2 s + 2 A B s + B^2$$

$$\text{or } A'^2 s^2 + 2B's + C' = 0 \quad \begin{aligned} A' &= A^2 - 1 \\ B' &= AB + \mu_o h \\ C' &= B^2 - h^2 \end{aligned}$$

Thus

$$s = \frac{-B' + \sqrt{B'^2 - A'C'}}{A'} \quad (8.51)$$

$$p^*(s) = \Sigma_{\uparrow}^{\vee}(s) \left(a - \beta \frac{\mu_o h - s}{A s + B} \right) (1 - \xi C)$$

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

$$u^*(s) = \Sigma_{\uparrow}^{\vee}(s) \Delta \Omega(s) \exp \left[- \int_0^s \Sigma_{\uparrow}^{\vee}(s') [a - \beta \mu(s')] ds' \right] \quad (8.52)$$

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:

$$u^*(s) = A_i \exp [-\alpha_i (s - s_i)] \quad \text{for } s_i < s < s_{i+1} \quad (8.53)$$

where

$$u^*(s_i) = \Sigma_{\uparrow, i}^{\vee} \Delta \Omega(s_i) \exp \left[- \int_0^{s_i} \Sigma_{\uparrow}^{\vee}(s') [a - \beta \mu(s')] ds' \right]$$

$$u^*(s_{i+1}) = \Sigma_{\uparrow, i+1}^{\vee} \Delta \Omega(s_{i+1}) \exp \left[- \int_0^{s_{i+1}} \Sigma_{\uparrow}^{\vee}(s') [a - \beta \mu(s')] ds' \right]$$

i.e.,

$$A_i = u^*(s_i)$$

$$\alpha_i = -\frac{1}{s_{i+1}-s_i} \ln \left[\frac{u^*(s_{i+1})}{u^*(s_i)} \right]$$

$$= \alpha \sum_{i=1}^N \frac{1}{s_{i+1}-s_i} \left[\ln \frac{\Delta \Omega(s_{i+1})}{\Delta \Omega(s_i)} - \beta \sum_{i=1}^N (s_{i+1}-s_i) \right] \quad (8.54)$$

The fractional solid angle is computed for the detector sphere in surface and volume flux calculations as:

$$\Delta \Omega(s_i) = 1 \text{ if } t_i < R_p$$

$$= \frac{1}{4} \left(\frac{R_p}{t_i} \right)^2 \left[1 + \frac{1}{4} \left(\frac{R_p}{t_i} \right)^2 \right] \text{ if } t_i > R_p \quad (8.55)$$

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:

$$\xi = \frac{\int_0^s u^*(s') ds'}{\int_0^s u^*(s') ds'} = \frac{1}{C} \left\{ \sum_{i=1}^{i-1} \frac{A_i}{\alpha_i} \left[1 - \exp \left[-\alpha_i \Delta s_i \right] \right] + \int_{s_i}^s A_i \exp \left[-\alpha_i (s'-s_i) \right] ds' \right\}$$

or

$$A_i \exp \left[-\alpha_i (s - s_i) \right] = A_i - \alpha_i \left\{ \xi C - \sum_{i=1}^{i-1} \frac{A_i}{\alpha_i} \left[1 - \exp(-\alpha_i \Delta s_i) \right] \right\} = D$$

where D is the right side of the above equation and C is the denominator of the original.

Thus $u^*(s) = D/C$

$$\text{and } s = s_i - \frac{1}{\alpha_i} \ln (D/A_i) \quad (8.56)$$

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 ($IN1 \leq 0$), 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 to the number of these comment cards, i.e. $IN1 = \text{total number of comment cards}$.

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.

<u>Column</u>	<u>Format</u>	<u>Symbol</u>	<u>Definition</u>
1-3	I3	IN1	input control for card 1-1 (descriptive cards) omit card 1-1 if $IN1 \leq 0$ supply IN1 physical cards if $IN1 \geq 1$
4-6	I3	IN2	input control for card 1-2 (first title card) omit card 1-2 if $IN2 \leq 0$ supply card 1-2 if $IN2 \geq 1$
7-9	I3	IN3	input control for card 1-3 (second title card) omit card 1-3 if $IN3 \leq 0$ supply card 1-3 if $IN3 \geq 1$
10-12	I3	IN4	input control for card 1-4 (geometric limits) omit card 1-4 if $IN4 \leq 0$ supply card 1-4 if $IN4 \geq 1$
13-15	I3	IN5	input control for card 1-5 (fixed source limits) omit card 1-5 if $IN5 \leq 0$ supply card 1-5 if $IN5 \geq 1$
16-18	I3	IN6	input control for card 1-6 (cross section limits) omit card 1-6 if $IN6 \leq 0$ supply card 1-6 if $IN6 \geq 1$
19-21	I3	IN7	input control for card 1-7 (neutron scattering limits) omit card 1-7 if $IN7 \leq 0$ supply card 1-7 if $IN7 \geq 1$
22-24	I3	IN8	input control for card 1-8 (flux limits) omit card 1-8 if $IN8 \leq 0$ supply card 1-8 if $IN8 \geq 1$
25-27	I3	IN9	input control for card 1-9 (sampling options) omit card 1-9 if $IN9 \leq 0$ supply card 1-9 if $IN9 \geq 1$

Column	Format	Symbol	Definition
28-30	I3	IN10	input control for card 1-10 (iteration limits) omit card 1-10 if $IN10 \leq 0$ supply card 1-10 if $IN10 \geq 1$
31-33	I3	IN11	printout control for storage map printout no printout of storage map if $IN11 \leq 0$ printout of storage map if $IN11 \geq 1$
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 $IN1 \leq 0$
b) Supply $IN1$ physical cards if $IN1 \geq 1$

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 $IN2 \leq 0$
b) Supply this card if $IN2 \geq 1$

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 $IN3 \leq 0$
b) Supply this card if $IN3 \geq 1$

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 $IN4 \leq 0$
b) Supply this card if $IN4 \geq 1$

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 $2 \cdot NSMAX - (\text{number of plane surfaces} + 1)$
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 $IN5 \leq 0$
b) Supply this card if $IN5 \geq 1$

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 $IN6 \geq 1$.

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.

1-3	I3	NORDER	Maximum number of elastic scattering transfer matrices for all elements. 1 for P_0 (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 $NORDER \geq 1$ 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_0 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 $IN8 \leq 0$.
b) Supply this card if $IN8 \geq 1$.

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	NUMAX	Number of Legendre moments of the angular flux 0, P_0 moment (always obtained, $\equiv$ scalar flux). 1, P_0 and P_1 moments, etc.
16-18	I3	NTMAX	Number of length-of-flight moments of the flux. 0, zeroth moment (always obtained, $\equiv$ 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 $IN9 \geq 1$.
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	MODEL P	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	MODEL Q	0, randomly select angular source variables in the source coordinate system. 1, randomly select angular source variables like MODEL V 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	MODEL U	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	MODEL V	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	1913	---	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 $IN10 \leq 0$.
b) Supply this card if $IN10 \geq 1$.

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	2013	---	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 $IN1 \leq 0$. Supply IN1 physical cards if $IN1 \geq 1$.
4-6	I3	IN2	Input control for Card 2-2 (surfaces). Omit Card 2-2 if $IN2 \leq 0$. Supply IN2 physical cards if $IN2 \geq 1$, i.e. IN2 surfaces will be described.
7-9	I3	IN3	Input control for Card 2-3 (regions). Omit Card 2-3 if $IN3 \leq 0$. Supply IN3 physical cards if $IN3 \geq 1$, 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	1913	---	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 $IN1 \leq 0$.
b) Supply $IN1$ physical cards if $IN1 \geq 1$.

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 $IN2 \leq 0$.
b) Supply $IN2$ physical cards if $IN2 \geq 1$.

1-3	I3	I	Index of the surface being described.
4-6	I3	NTP(I)	Index (i) 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. $1 \leq NEX \leq 13$, 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.
28-36	E9.0	AA(3)	Third parameter defining the surface.
37-45	E9.0	AA(4)	Fourth parameter defining the surface.
46-54	E9.0	AA(5)	Fifth parameter defining the surface.
55-63	E9.0	AA(6)	Sixth 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
quadratic	$a_0 + a_1x + a_2y + a_3z + a_4x^2 + a_5xy + a_6xz + a_7y^2 + a_8yz + a_9z^2 = 0$	i (input)	0	$a_0, a_1, \dots, a_9$
plane I x-axis	$x = c$	1	1	c
plane I y-axis	$y = c$	2	2	c
plane I z-axis	$z = c$	3	3	c
plane II x-axis	$(y - y_0)/(z - z_0) = (y_1 - y_0)/(z_1 - z_0)$	3	4	y_0, y_1, z_1, z_0
plane II y-axis	$(x - x_0)/(z - z_0) = (x_1 - x_0)/(z_1 - z_0)$	3	5	x_0, x_1, z_1, z_0
plane II 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 II x-axis	$\left[\frac{(y - y_0)^2 + (z - z_0)^2}{(y_1 - y_0)^2 + (z_1 - z_0)^2} \right]^{1/2} = \frac{(x - x_0)}{(x_1 - x_0)}$	6	7	$y_0, z_0, y_1, z_1, x_1, x_0$
cone II y-axis	$\left[\frac{(x - x_0)^2 + (z - z_0)^2}{(x_1 - x_0)^2 + (z_1 - z_0)^2} \right]^{1/2} = \frac{(y - y_0)}{(y_1 - y_0)}$	6	8	$x_0, z_0, x_1, z_1, y_1, y_0$
cone II z-axis	$\left[\frac{(x - x_0)^2 + (y - y_0)^2}{(x_1 - x_0)^2 + (y_1 - y_0)^2} \right]^{1/2} = \frac{(z - z_0)}{(z_1 - z_0)}$	6	9	$x_0, y_0, x_1, y_1, z_1, z_0$
cylinder II x-axis	$(y - y_0)^2 / a^2 + (z - z_0)^2 / b^2 = 1$	6	10	y_0, a, z_0, b
cylinder II y-axis	$(x - x_0)^2 / a^2 + (z - z_0)^2 / b^2 = 1$	6	11	x_0, a, z_0, b
cylinder II 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

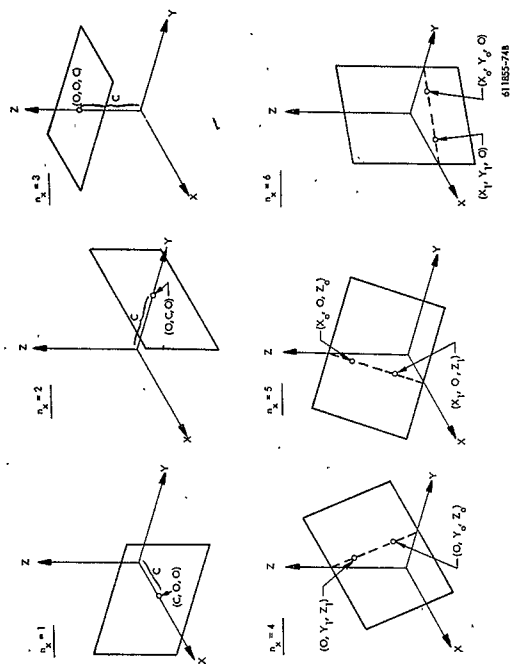


Figure 7. Plane Surfaces

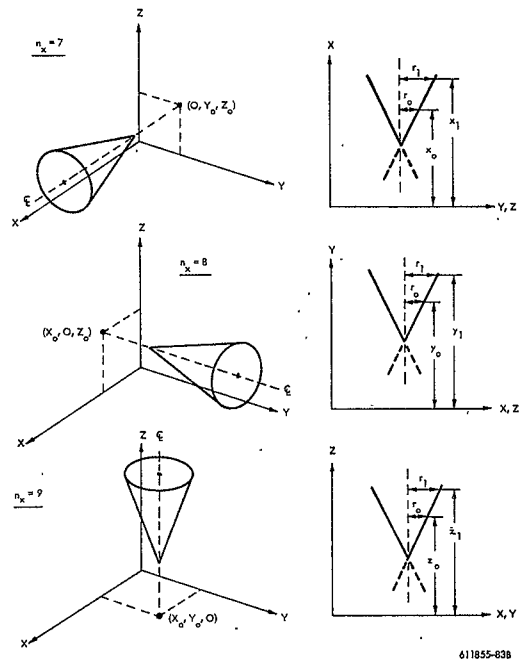


Figure 8. Conical Surfaces

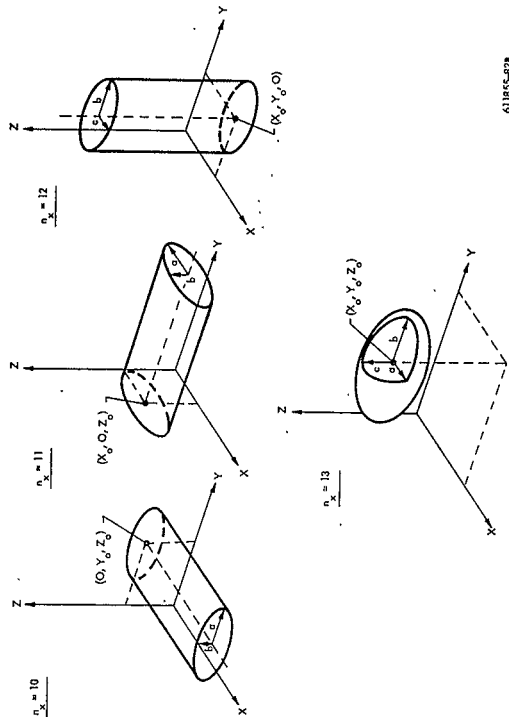


Figure 9. Elliptical Cylinders and Ellipsoids

61855-928

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 $IN3 \leq 0$.
b) Supply $IN3$ physical cards if $IN3 \geq 1$.

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 $MODEL P = 1$. 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).

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.			
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.

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 $IN2 \leq 0$.

b) Supply this card(s) if $IN2 \geq 1$.

c) Source spectra can be input in any desired group structure and will be regrouped to this set of groups.

1-72	BE9.0	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 $IN3 \leq 0$.

b) Supply this card, and Cards 3-4 and 3-5 as required, for $IN3$ sources if $IN3 \geq 1$.

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^1 (radians)	θ^1 (radians)	θ^1 (radians)
J = 5	μ^1	μ^1	μ^1

Azimuthal angles are in the range $-\pi \leq \theta, \theta^1 \leq \pi$

Cosines of polar angles are in the range $-1 \leq \mu, \mu^1 \leq 1$

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 $I^1 = -NPC(J, I)$ is used and Card 3-4 is not required.

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 $I^1 = -MAX$.
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 $I^1 = -MAX$ has been normalized to the total source strength).

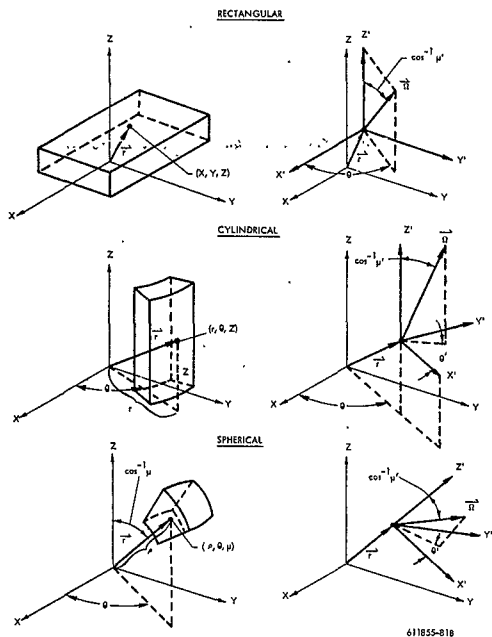


Figure 10. Source Distribution Variables

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:
- 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.
 - Omit for any variable for which NPC(J, I) < 0.
 - This distribution is normalized and interpolated linearly.
 - Histogram data can be used (points can coincide).

Column	Format	Symbol	Definition
1-72	8E9.0	VEE(1,J,I)	Minimum value of the J^{th} 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 J^{th} variable at its minimum value (not used if the variable is discrete, NPC(J,I) = 1)
		.	
		.	
		.	
		VEE(K,J,I)	Kth value of the J^{th} 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 J^{th} 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:
- 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 K^{th} energy point $E(K)$. The units of $EN(K)$ are particles/Mev if $ISP = 0$ or Mev/Mev if $ISP = 1$.
 - If $IAL = 0$, the input on Card 3-5 consists of alternating values of energy and spectrum
 $E(1), EN(1), E(2), EN(2), \dots, E(MAX), EN(MAX)$
 - 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
 $EN(1), EN(2), \dots, EN(MAX)$
 - If $IAL = 2$, the energy points are defined first
 $E(1), E(2), \dots, E(MAX)$
 and then followed by another card with the corresponding spectrum
 $EN(1), EN(2), \dots, EN(MAX)$
 - 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 K^{th} 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 group-wise 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 $IN4 \leq 0$.
b) Supply this card(s) if $IN4 \geq 1$ with source indices for $IN4$ regions.
c) This data is required if and only if $MODEL P=1$.
d) This data can also be input on Card 2-3.

1-72	2413	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	
		I	Second geometric region index	K = 2
		ISV(I)	Index of source superimposed over Region I (source which completely covers the region)	
		⋮	⋮	⋮
		I	Last geometric region index	K = IN4
		ISV(I)	Index of source superimposed over Region I (source which completely covers the region)	
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 $IN1 \leq 0$. Supply IN1 physical cards if $IN1 \geq 1$
4-6	I3	IN2	Input control for Card 4-2 (neutron scattered energies). Omit Card 4-2 if $IN2 \leq 0$. Supply Card 4-2 if $IN2 \geq 1$.
7-9	I3	IN3	Input control for microscopic cross sections; Omit Cards 4-3 through 4-7 if $IN3 \leq 0$; Supply Cards 4-3 through 4-7 as required if $IN3 \geq 1$.
10-12	I3	IN4	Input control for Card 4-8 (material-in-region) Omit Card 4-8 if $IN4 \leq 0$. Supply Card 4-8 with $IN4$ material-in-region indices if $IN4 \geq 1$.
13-15	I3	IN5	Input control for Card 4-9 (hydrogen in region) Omit Card 4-9 if $IN5 \leq 0$. Supply Card 4-9 with $IN5$ hydrogen-in-region densities if $IN5 \geq 1$.
16-18	I3	IN6	Cross section output option (used only if $IN3 \geq 1$), no output if $IN6 \leq 0$. If $IN6 \geq 1$, 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) $\times 10^{24}$ and then for each composite material in (Mev/cm ² /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 $IN1 \leq 0$.
b) Supply $IN1$ physical cards if $IN1 \geq 1$.

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 $IN2 \leq 0$.
b) Supply this card (s) if $IN2 \geq 1$.
c) This data is required if and only if $NXSECT=1$ (neutron problem)

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 $IN3 \leq 0$.
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 $IN3 \geq 1$.
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 material 1, 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.

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	2413	LMAX	$ LMAX $, number of elastic scattering transfer matrices. 1 for P_0 only 2 for P_0 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.

Column	Format	Symbol	Definition
		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.

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.

1-72	8E9.0	$\sigma_{j \rightarrow k}^l$	(1) l th Legendre moment of the transfer
		XSE(J, K, L)	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_0 in-group transfer for all energy groups

$$\sigma_{i \rightarrow j}^0, i = 1, 2, \dots, NEMAX$$

Start a new card with P_0 down 1 transfer for all groups except the last

$$\sigma_{i \rightarrow i+1}^{ne}, i = 1, 2, \dots, NEMAX - 1$$

•
•

Start a new card with P_0 down (NDSM-1) transfer

$$\sigma_{i \rightarrow j + NDSM - 1}^{ne}, i = 1, 2, \dots, NEMAX - (NDSM-1)$$

Start a new card with P_1 in group transfer

$$\sigma_{i \rightarrow i}^{ne}, i = 1, 2, \dots, NEMAX$$

•
•

Start a new card with $P_{\lfloor LMAX \rfloor - 1}$ down (NDSM-1) transfer

$$\sigma_{i \rightarrow j + NDSM - 1}^{ne}, i = 1, 2, \dots, NEMAX - (NDSM-1)$$

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	$\sigma_{i \rightarrow i + k - 1}^{ne}$	(1) Non-elastic transfer coefficient from group i to group $i + k - 1$
		XSI(K, J)	$= \sigma_{i \rightarrow i + k - 1}^{inelastic} + 2 \sigma_{i \rightarrow i + k - 1}^{n-2n} + \dots$
73-80	2A4	----	Any desired information for card identification.

(1) Start the first card with non-elastic transfer from group 1

$$\sigma_{i \rightarrow i + K - 1}^{ne}, K = 1, 2, \dots, KMX(1)$$

Start a new card with non-elastic transfer from group 2

$$\sigma_{2 \rightarrow 2 + K - 1}^{ne}, K = 1, 2, \dots, KMX(2)$$

•
•

Start a new card with non-elastic transfer from group JMAX

$$\sigma_{JMAX \rightarrow JMAX + K - 1}^{ne}, K = 1, 2, \dots, KMX(JMAX)$$

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
		MTL(I)	>0, index of material in this region
			= 0, region contains hydrogen only
			< 0, region is void
		I	Second region index
		MTL(I)	Material index for region
		• • •	• • •
		I	Last region index
		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.

1-72	6(13, E9, 0)	I	First Region Index	} K=1
		RHO(I)	Hydrogen density in region I units according to NUNJTD	
		I	Second region index	} K=2
		RHO(I)	Hydrogen density in region I	
		.	.	.
		.	.	.
		.	.	.
		I	Last region index	} K=IN5
		RHO(I)	Hydrogen density in region I	
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.

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 $IN4$ detectors 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 $IN1 \geq 1$.

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.

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 $IN3 \leq 0$. b) Supply this card for $IN3$ response functions if $IN3 \geq 1$.			
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 $\text{cm}^{-2} \text{sec}^{-1}$). 1, energy flux response input by flux group boundary with units (response/Mev $\text{cm}^{-2} \text{sec}^{-1}$) <0, energy deposition response function for region I' = -NTP with units (Mev $\text{cm}^{-3} \text{sec}^{-1}$ /particle $\text{cm}^{-2} \text{sec}^{-1}$). 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(I, 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 .			
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 $IN4 \leq 0$. b) Supply $IN4$ physical cards if $IN4 \geq 1$.			
1-3	I3	I	Index of the detector being described.
4-6	I3	IDR(I)	0, for point detector >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 $IN5 \leq 0$.
b) Supply this card if $IN5 \geq 1$.

1-72	24I3	ISV(1)	Index of first source for which the individual scalar flux contribution is required.
		$\vdots$	
		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 $IN6 \leq 0$
b) Supply this card if $IN6 \geq 1$

Column	Format	Symbol	Definition
1-72	24I3	ISR(1)	Index of the first non-vaid region from which the individual scattered scalar flux contribution is required.
		$\vdots$	
		ISR(NSRMAX)	Index of the last non-vaid 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.

1-3	I3	IN1	Input control for Card 6-1 (descriptive cards) Omit Card 6-1 if $IN1 \leq 0$. Supply, IN1 physical cards if $IN1 \geq 1$.
4-6	I3	IN2	Input control for Card 6-2 (spherical source and detector) Omit Card 6-2 if $IN2 \leq 0$. Supply Card 6-2 if $IN2 \geq 1$.
7-9	I3	IN3	Input control for Card 6-3 (source importance). Omit Card 6-3 if $IN3 \leq 0$. Supply Card 6-3 if $IN3 \geq 1$.
10-12	I3	IN4	Input control for Card 6-4 (source variable importance). Omit Card 6-4 if $IN4 \leq 0$. Supply Card 6-4 if $IN4 \geq 1$.
13-15	I3	IN5	Input control for Card 6-5 (group importance). Omit Card 6-5 if $IN5 \leq 0$. Supply Card 6-5 if $IN5 \geq 1$.
16-18	I3	IN6	Input control for Card 6-6 (linear buildup). Omit Card 6-6 if $IN6 \leq 0$. Supply Card 6-6 if $IN6 \geq 1$.

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 $IN1 \geq 1$.

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 $IN2 \geq 1$.

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, $MODEL P = 0$ and more than one source is present.

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 $MODEL P = 0$.
d) The last two pieces of data are required if $MODEL Q = 0$.

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 $IN4 \geq 1$, i.e.,

Card 6-4 } Source 1
Card 6-4' }
⋮
Card 6-4 } Last Source
Card 6-4' }

- c) All numbers on this card must be greater than 0.0.

1-9	E9.0	ALP(1, I)	Relative importance of the preferred value of the first source variable for the <u>I</u> th 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 <u>I</u> th 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 $IN5 \geq 1$.

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 $IN6 \geq 1$.

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 $IN7 \geq 0$.
c) All numbers on this card must be greater than 0.0.

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 $\frac{d\Sigma}{d\Omega} (0^\circ \text{ scatter}) \times \text{energy after scatter } (0^\circ)$ $\frac{d\Sigma}{d\Omega} (180^\circ \text{ scatter}) \times \text{energy after scatter } (180^\circ)$ using the Klein-Nishina formula for an average group energy has yielded good results.			
<u>CARD 6-8, Hydrogen Scattering Importance</u>			
			NOTE: a) Omit this card if $IN8 < 0$. b) Supply this card if $IN8 \geq 1$. c) All numbers on this card must be greater than 0.0.
1-72	8E9.0	ALH(1) : : : ALH(NEMAX)	Ratio of forward-to-backward scattering importance for hydrogen for the first energy group. (See notes below.) 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., 10^5 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 $IN9 < 0$. 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 importance 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

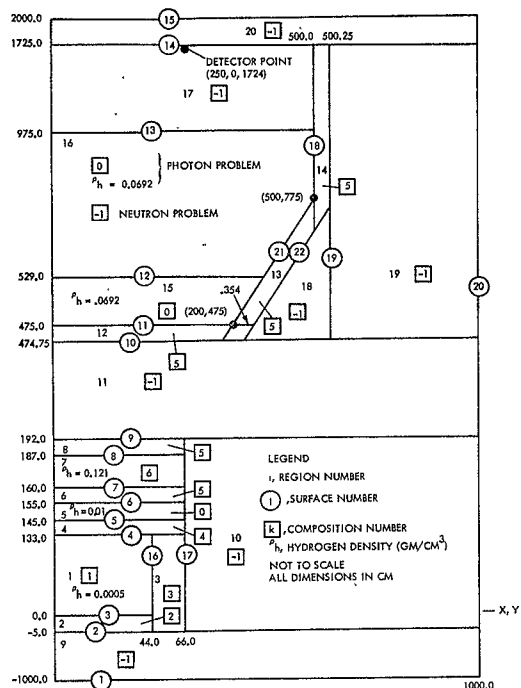


Figure 11. Sample Problem Geometry

611855-758

TABLE 2

THE SAMPLE PROBLEM PRINTOUT IS PRESENTED ON PAGES 154 THROUGH 169

ASTRACHAN
Laboratory

158

159

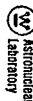
**Astronuclear
Laboratory**

[illegible]

SAMPLE PROBLEM 2: NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY EMPTY PROPELLANT TANK. REFERENCE ER-0236 (LOCKHEED STUDY FOR NASA-MSPC) *****THE FASTER CODE*****CASE 2 *****M.JORDAN,WAHL*****PASS

162

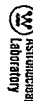
163



SAMPLE PROBLEM 2: NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY EMPTY PROPELLANT TANK. REFERENCE ER-0236 (LOCKHEED STUDY FOR NASA-MSPC) *****THE FASTER CODE*****CASE 2 *****M.JORDAN,WAHL*****PASS

162

163



[illegible]

SAMPLE PROBLEM 2: NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY *****THE PASTER CODE*****CASE 2
 EMPTY PROPELLANT TANK. REFERENCE EN=8236 (LOCKHEED STUDY FOR NASA=MSFC) *****J. JORDAN/WALL*****PAGE 6



*****FLUXES FOR DETECTOR 1 AFTER 400 PACKETS*****
 CALCULATED AVERAGE NUMBER FLUX ENERGY FLUX NUMBER FLUX ENERGY FLUX
 PRECISION ENERGY-MEV THIS GROUP THIS GROUP DERIVATIVE DERIVATIVE CUMULATIVE CUMULATIVE
 GROUP 1... 1.2051E+01 8.6185E+00 2.1542E+06 1.8556E+07 6.7388E+05 8.3932E+06 2.1542E+06 1.8556E+07
 GROUP 2... 2.2114E+01 1.6958E+00 3.3151E+06 3.5306E+07 6.3540E+06 6.1823E+07 4.3517E+07 6.1823E+07
 GROUP 3... 3.8512E+01 4.7102E+00 4.0302E+07 1.9007E+08 1.0891E+07 7.9806E+07 5.1829E+07 2.7394E+08
 GROUP 4... 3.8927E+01 3.2740E+00 2.0397E+07 6.6790E+07 2.5057E+07 8.2039E+07 7.2228E+07 3.4074E+08
 GROUP 5... 5.1440E+01 3.5457E+00 2.0871E+07 3.0154E+08 2.2888E+07 6.0022E+08 1.2101E+08 4.9421E+08
 GROUP 6... 6.1307E+01 1.4103E+00 3.6454E+08 5.1630E+08 3.0972E+08 4.3866E+08 4.7684E+08 9.5833E+08
 GROUP 7... 7.8850E+01 9.930E+01 3.9528E+07 3.7108E+07 1.7008E+08 1.5961E+08 5.1624E+08 9.9581E+08
 GROUP 8... 8.1606E+01 7.0911E+01 1.9277E+07 5.4079E+07 4.3559E+08 3.4923E+08 5.3554E+08 1.0212E+09
 GROUP 9... 9.6021E+01 5.2825E+01 8.3204E+07 4.3953E+07 3.3149E+08 1.7511E+08 6.7875E+08 1.0509E+09
 GROUP 10... 10.3452E+01 2.9145E+00 3.9344E+07 3.9344E+07 6.4607E+08 3.6121E+08 7.8165E+08 1.1271E+09
 GROUP 11... 11.5626E+01 2.4982E+01 2.8943E+07 7.2303E+06 2.4321E+08 6.0759E+07 7.9505E+08 1.1343E+09
 GROUP 12... 12.6946E+01 1.6553E+01 1.7252E+07 2.8957E+06 2.3961E+08 3.9462E+07 8.0784E+08 1.1372E+09
 GROUP 13... 13.1294E+01 1.0244E+01 8.0437E+07 4.7940E+07 4.8783E+08 6.0822E+08 1.1421E+09
 *****NUMBER FLUX RESPONSES FOR DETECTOR 1 AFTER 400 PACKETS*****
 RAD(TTS)/HR RAD(ETH)/HR REM/HOUR
 GROUP 1... 1.6775E+01 5.6133E+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... 2.5388E+02 4.1316E+02 2.0514E+03
 GROUP 5... 4.8044E+02 7.0717E+02 3.7877E+03
 GROUP 6... 3.4386E+03 4.6691E+03 3.3707E+04
 GROUP 7... 3.1261E+02 4.2232E+02 3.2526E+03
 GROUP 8... 5.2246E+02 7.6438E+02 5.4310E+03
 GROUP 9... 4.7915E+02 6.5010E+02 5.8044E+03
 GROUP 10... 4.1572E+02 5.9248E+02 4.2733E+03
 GROUP 11... 1.0816E+02 1.5771E+02 1.0689E+03
 GROUP 12... 5.8800E+01 8.0399E+01 1.508E+02
 GROUP 13... 1.0205E+02 1.4695E+02 8.5719E+02
 TOTALS... 7.0624E+03 9.8050E+03 6.5348E+04
 MIN ERROR... 3.1400E+01 3.0951E+01 3.2971E+01
 MAX ERROR... 5.8951E+01 8.5575E+01 5.9727E+01
 *****NUMBER FLUX MOMENTS FOR DETECTOR 1 AFTER 400 PACKETS*****
 TYPANT 1 REGION 1 REGION 2 REGION 3 REGION 4 REGION 5
 GROUP 1... 2.1542E+06 1.8556E+07 0. 6.7903E+04 3.4293E+04 8.0104E+01 3.4430E+01 3.1488E+05
 GROUP 2... 9.3731E+06 2.5306E+05 0. 3.1241E+05 6.0207E+04 6.1888E+02 4.1745E+04 5.2598E+05
 GROUP 3... 4.0302E+07 1.9007E+08 0. 1.0401E+06 4.4902E+05 1.4927E+04 6.3746E+04 1.0998E+08
 GROUP 4... 2.0397E+07 6.6790E+07 0. 7.7371E+05 8.1115E+04 1.3634E+02 8.0814E+03 8.8497E+04

166

SAMPLE PROBLEM 2: NEUTRON DOSE RATE AT POINT DETECTOR ABOVE PARTIALLY *****THE PASTER CODE*****CASE 2
 EMPTY PROPELLANT TANK. REFERENCE EN=8236 (LOCKHEED STUDY FOR NASA=MSFC) *****J. JORDAN/WALL*****PAGE 7

*****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... 1.8301E+04 2.7857E+03 1.4649E+03 8.0038E+04 6.6346E+05 8.8705E+05 7.3522E+05 9.0527E+05
 GROUP 2... 2.8377E+04 1.1266E+04 4.0191E+04 1.0177E+06 6.3954E+06 6.1623E+05 4.8952E+06 1.5835E+06
 GROUP 3... 2.8657E+04 3.6268E+03 1.0526E+06 1.2189E+07 2.4489E+07 1.1771E+05 1.0948E+07 5.0170E+06
 GROUP 4... 6.0821E+03 8.3508E+02 1.6573E+06 1.1542E+07 6.2324E+06 1.0202E+04 1.2886E+05 1.7964E+06
 GROUP 5... 7.1301E+03 2.2848E+02 2.2848E+06 3.1102E+06 3.5086E+06 5.5005E+03 2.6451E+05 1.9470E+06
 GROUP 6... 3.4900E+03 9.8509E+01 3.0693E+07 1.2204E+08 2.1102E+08 1.8641E+07 7.9907E+05 3.2742E+06
 GROUP 7... 1.0110E+03 1.9263E+06 6.4515E+06 2.9411E+06 1.6037E+06 6.1613E+01 2.0155E+04 3.9441E+05
 GROUP 8... 2.6381E+01 3.2700E+01 1.0141E+07 2.4345E+07 4.8701E+07 2.0950E+02 2.9579E+04 3.0962E+05
 GROUP 9... 3.6846E+06 3.4893E+02 1.2925E+07 2.3297E+07 4.6986E+07 5.9251E+05 3.4275E+04 3.2087E+05
 GROUP 10... 6.4981E+01 1.0495E+05 5.0985E+05 7.0392E+05 4.4951E+05 6.4463E+03 1.2820E+05
 GROUP 11... 4.2501E+03 2.6677E+05 6.3432E+06 1.6457E+06 1.6325E+07 6.6324E+11 1.0569E+04 6.2457E+05
 GROUP 12... 1.0756E+05 5.4446E+08 5.5212E+06 3.1613E+06 1.0569E+07 3.7495E+05 2.2042E+03 6.0539E+04
 GROUP 13... 1.2404E+04 1.9312E+06 6.2045E+06 1.8183E+06 1.4746E+06 5.0788E+04 1.7874E+07 2.3111E+07
 RAD(TTS)/HR 1.5001E+00 2.5096E+01 6.6248E+02 2.8970E+03 5.0844E+03 2.8716E+01 2.6214E+02 2.6128E+02
 RAD(ETH)/HR 1.1448E+00 3.6285E+01 9.2271E+02 3.6095E+03 5.0844E+03 2.8716E+01 2.6214E+02 2.6128E+02
 REM/HOUR 1.0110E+01 1.6445E+01 3.3853E+03 4.3657E+04 3.4355E+04 1.8448E+02 1.9064E+03 1.9064E+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... 1.4191E+06 3.1786E+04 2.1142E+03 2.5837E+02 1.9107E+01 1.0572E+02 2.1104E+02 2.1247E+02
 GROUP 2... 1.5214E+06 1.0852E+06 3.4913E+04 2.2811E+04 1.8854E+03 1.2013E+03 9.8926E+02 4.6004E+04
 GROUP 3... 3.9308E+06 3.7396E+06 1.7153E+06 2.2221E+06 4.6816E+05 1.8814E+06 3.7118E+07 9.3244E+07
 GROUP 4... 6.4312E+06 1.9312E+06 1.1831E+06 5.3219E+06 1.4746E+06 5.0788E+04 1.7874E+07 2.3111E+07
 GROUP 5... 1.7303E+07 2.2565E+06 8.1941E+05 1.5129E+07 8.6028E+05 2.8532E+05 3.3925E+07 4.6540E+07
 GROUP 6... 5.6557E+06 6.5280E+05 8.3846E+05 2.0702E+06 4.6733E+06 9.7477E+05 3.2444E+08 4.1287E+08
 GROUP 7... 9.5559E+01 1.3853E+06 4.4949E+05 2.5308E+07 1.7633E+01 1.4903E+05 3.4446E+07 4.4038E+07
 GROUP 8... 6.3669E+06 9.8255E+05 2.9981E+05 2.3610E+07 2.1018E+06 2.5023E+05 7.4389E+07 8.9099E+07

167



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) *****M. JORDAN, WASH*****PAGE 8

GROUP 9...	6.1157E+06	1.1127E+06	5.2258E+05	2.1237E+07	2.7156E+06	3.9784E+05	7.8164E+07	9.4681E+07
GROUP 10...	1.7787E+06	3.1580E+05	1.3075E+05	6.4883E+06	7.5750E+05	1.1698E+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...	7.0019E+05	9.5088E+05	1.1987E+05	1.4309E+06	2.9012E+05	5.9487E+04	1.6455E+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(TTS)/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	1.8421E+03	4.2143E+02	1.3774E+02	2.2738E+03	2.0926E+02	6.7597E+01	9.0401E+03	1.1049E+04
RM/NOUR	1.0392E+04	2.4440E+03	6.9632E+02	1.5578E+04	1.3835E+03	3.9158E+02	6.0355E+04	7.3915E+04



168

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) *****M. JORDAN, WASH*****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)
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)
REGION 11	(-9, 10)	(10, 12)	(20, 31)
REGION 12	(-10, 11)	(11, 15)	(-21, 0)
REGION 13	(-10, 0)	(18, 14)	(21, 16)
REGION 14	(14, 0)	(-18, 17)	(19, 19)
REGION 15	(-11, 12)	(12, 16)	(-21, 13)
REGION 16	(-12, 15)	(13, 17)	(18, 14)
REGION 17	(-13, 16)	(14, 20)	(18, 14)
REGION 18	(-10, 11)	(19, 19)	(22, 13)
REGION 19	(-19, 0)	(20, 31)	(-10, 11)
REGION 20	(-14, 0)	(15, 31)	(20, 31)

169



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/cm² 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/cm² 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:

- total flux by source, labeled SOURCE XXX (not used in this problem since only 1 source was present)
- total scattered flux by scattering region, labeled REGION XXX (obtained in this problem for every non-vold region)
- total flux by order-of-scatter, labeled SCATTER XXX (obtained through the 7th scatter for this problem)
- Legendre moments of the angular flux, labeled ANGULAR XXX (the current was obtained for this problem)
- 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-region). 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 ₂ O) $\sim 6.6 \times 10^2$ (QAD, C)

*Reference 20, Figures 30 and 55

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. Joannou and J. S. Dudek, "GAM-1: A Consistent P₁ 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 Computational 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

$$\phi_0(\vec{r}, E) = \iint_{4\pi} \left\{ \frac{\phi_0(\vec{r}, \vec{\Omega}, E)}{q(\vec{\Omega})} \right\} q^*(\vec{\Omega}) d\Omega \quad (A.1)$$

$$\approx \frac{1}{n} \sum_{i=1}^n \phi_0^*(\vec{r}, \vec{\Omega}_i, E) \quad (A.2)$$

where

$$\phi_0^*(\vec{r}, \vec{\Omega}_i, E) = \frac{1}{q(\vec{\Omega}_i)} \phi_0(\vec{r}, \vec{\Omega}_i, E) \quad (A.3)$$

$$\vec{\Omega}_i \text{ selected at random from } q^*(\vec{\Omega}) \quad (A.4)$$

$$\left. \begin{aligned} q^*(\vec{\Omega}) &\geq 0 \\ q^*(\vec{\Omega}) &> 0 \text{ if } \int_0^\infty \phi_0(\vec{r}, \vec{\Omega}, E) dE > 0 \end{aligned} \right\} \quad (A.5)$$

$$\iint_{4\pi} q^*(\vec{\Omega}) d\Omega = 1$$

Thus the angular flux is approximated for finite n by:

$$\phi_0^*(\vec{r}, \vec{\Omega}_1, E) = \frac{1}{n} \sum_{i=1}^n \phi_0^*(\vec{r}, \vec{\Omega}_i, E) \delta(\vec{\Omega} - \vec{\Omega}_i) \quad (A.6)$$

But $\phi_0^*(\vec{r}, \vec{\Omega}_1, E)$ is given by a simple line integral, i.e., equation 2.2:

$$\phi_0^*(\vec{r}, \vec{\Omega}_1, E) = \frac{1}{q(\vec{\Omega}_1)} \int_0^{\infty} S_0(\vec{r} - s\vec{\Omega}_1, \vec{\Omega}_1, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}_1, E) ds'\right] ds \quad (A.7)$$

This integration can be performed numerically with extreme accuracy for non-vold source regions with slowly varying source distributions by using the transformations:

$$du = \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}_1, E) ds'\right] ds \quad (A.8)$$

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:

$$\phi_0^*(\vec{r}, \vec{\Omega}_1, E) = \frac{1}{q(\vec{\Omega}_1)} \int_0^{\infty} S_0(\vec{r} - s(u)\vec{\Omega}_1, \vec{\Omega}_1, E) \exp\left[-\int_0^{s(u)} \left\{ \Sigma^t(\vec{r} - s'\vec{\Omega}_1, \vec{E}) - \Sigma^t(\vec{r} - s'\vec{\Omega}_1, \bar{E}) \right\} ds'\right] du \quad (A.9)$$

where $s(u)$ is the solution for s of

$$u = \int_0^{s(u)} \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}_1, \bar{E}) ds'\right] ds \quad (A.10)$$

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.)

$$\phi_1^*(\vec{r}, E) = \iiint \left\{ \frac{S_1(\vec{r}, \vec{\Omega}, E) \exp\left[-\int_0^s \Sigma^t(\vec{r} - s'\vec{\Omega}, E) ds'\right]}{p_1^*(\vec{r}') \cdot s^2} \right\} p_1^*(\vec{r}') dV \quad (A.11)$$

which yields a point, single-scattered source

$$W_1^s(\vec{\Omega}, E) = \frac{1}{p_1^*(\vec{r}_1)} S_1(\vec{r}_1, \vec{\Omega}, E) \quad (A.12)$$

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, $W_0(\vec{\Omega}, E)$, at $\vec{r}_0$ which finally yielded $W_1^s(\vec{\Omega}, E)$. The point source $W_0(\vec{\Omega}, E)$ 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.

$$W_1^*(\vec{n}, E) = \frac{1}{p_1(\vec{r}_1)} \iint_{4\pi} \left(\frac{\int_0^\infty \phi_0(\vec{r}_1, \vec{n}', E') \frac{d^2 \Sigma^s}{d\Omega dE}(\vec{r}_1; \vec{n}', E' \rightarrow \vec{n}, E) dE'}{q^*(\vec{n}')} \right) q^*(\vec{n}') d\Omega' \quad (A.13)$$

$$= \frac{1}{p_1(\vec{r}_1)} \frac{1}{q^*(\vec{n}_1)} \int_0^\infty \phi_0(\vec{r}_1, \vec{n}_1, E') \frac{d^2 \Sigma^s}{d\Omega dE}(\vec{r}_1; \vec{n}_1, E' \rightarrow \vec{n}, E) dE' \quad (A.14)$$

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 $q^*(\vec{n})$. 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:

$$\Delta \phi_{i,j,k}^*(\vec{n}) = \Delta \phi_{i,j,k} \delta(\vec{n} - \vec{n}_o)$$

where $\Delta \phi_{i,j,k}$ is the flux contribution to the i th energy group from the k th inner iteration of the i th outer iteration and is obtained for the fixed direction $\vec{n}_o$.

The angular moments are averaged over the azimuthal angle and obtained with respect to a preferred direction $\vec{n}_p$:

$$\phi_i^l = \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^\infty \iint_{4\pi} \Delta \phi_{i,j,k} \delta(\vec{n} - \vec{n}_o) p_l(\vec{n} \cdot \vec{n}_p) d\Omega \quad (B.1)$$

$$= \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^\infty \Delta \phi_{i,j,k} p_l(\vec{n}_o \cdot \vec{n}_p) \quad (B.2)$$

$$\left. \begin{aligned} P_0(\mu) &= 1 \\ P_1(\mu) &= \mu \\ P_l(\mu) &= \frac{1}{l!} \left[(2l-1) \mu P_{l-1}(\mu) - (l-1) P_{l-2}(\mu) \right] \end{aligned} \right\} \quad (B.3)$$

The preferred direction $\vec{n}_p$ is fixed by input for point and volume detectors. For surface detectors, it is the unit normal to the surface, $\vec{n}_s$, 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 $\vec{n}_p$.

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

$$\phi_l(\mu) = \sum_{l=0}^{\infty} \frac{2l+1}{4\pi} \phi_l^l P_l(\mu) \quad (B.4)$$

b) interval integrated flux, $\mu_m \leq \mu \leq \mu_{m+1}$

$$\bar{\phi}_{l,m} = \sum_{l=0}^{\infty} \frac{2l+1}{2} \phi_l^l \int_{\mu_m}^{\mu_{m+1}} P_l(\mu) d\mu \quad (B.5)$$

Using the relationship (Reference 15, page 309):

$$P_l(\mu) = \frac{1}{2^{l+1}} \left\{ \frac{d^l}{d\mu^l} P_{l+1}(\mu) - \frac{d^l}{d\mu^l} P_{l-1}(\mu) \right\} \quad (B.6)$$

Then

$$\bar{\phi}_{l,m} = \frac{1}{2} \sum_{l=0}^{\infty} \phi_l^l \left\{ P_{l+1}(\mu_{m+1}) - P_{l+1}(\mu_m) - P_{l-1}(\mu_{m+1}) + P_{l-1}(\mu_m) \right\} \quad (B.7)$$

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

$$\bar{\phi}_i^m = \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^{\infty} \Delta \phi_{i,k} \left(\frac{t_{i,k}}{t_o} \right)^m \quad (B.8)$$

$$t_o = \max \left(\left| \vec{r}_p - \vec{r}_c \right|, 100 \right), \quad \vec{r}_p, \vec{r}_c \text{ are defined in Section 8.3} \quad (B.9)$$

$$t_{i,k} = \sum_{k'=0}^k \left| \vec{r}_{i,k'} - \vec{r}_{i,k'-1} \right| + \Delta t_{i,k} \quad (B.10)$$

where

$$\begin{aligned} \Delta t_{i,k} &= \left| \vec{r} - \vec{r}_{i,k} \right| \text{ for a point detector} \\ &= s, \text{ the distance to a surface detector} \end{aligned} \quad (B.11)$$

Moments of the average length-of-flight are obtained for volume detectors.

$$\begin{aligned} \Delta t_{i,k} &= s + \Delta s/2 \text{ if void} \\ &= s + 1/\Sigma_i^v - \Delta s / [\exp(\Sigma_i^v \Delta s) - 1] \text{ if not void} \end{aligned} \quad (B.12)$$

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

$$\begin{aligned} \bar{\phi}_i^m &= \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^{\infty} \Delta \phi_{i,k} \tau_{i,k}^m \\ &= \frac{1}{n} \sum_{i=1}^n \sum_{k=0}^{\infty} \Delta \phi_{i,k} \left(\frac{t_{i,k}}{c} \right)^m \\ &= \left(\frac{t_o}{c} \right)^m \bar{\phi}_i^m \end{aligned} \quad (B.14)$$

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.

$$\begin{aligned} \phi_i(\tau) &= \exp \left[-a(\tau - \tau_o) \right] \sum_{m=0}^{\infty} b_{m'} (\tau - \tau_o)^{m'}, \tau \geq \tau_o \\ &= 0, \tau < \tau_o \end{aligned} \quad (B.15)$$

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

$$\begin{aligned}\hat{\phi}_i^m &= \int_{\tau_a}^{\infty} \phi_i(\tau) \tau^m d\tau \\ &= \int_{\tau_a}^{\infty} \exp[-\alpha(\tau - \tau_a)] \sum_{m'=0}^m b_{m'} (\tau - \tau_a)^{m'} \tau^m d\tau \\ &= \sum_{m'=0}^m b_{m'} c_{m,m'}, \quad m=0, 1, \dots\end{aligned}\quad (B.16)$$

where

$$c_{m,m'} = \int_{\tau_a}^{\infty} \exp[-\alpha(\tau - \tau_a)] \tau^m (\tau - \tau_a)^{m'} d\tau \quad (B.17)$$

Equation B.16 is solved for the constants $b_{m'}$, $m'=0, 1, \dots$ 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:

- Change of input tape logical designation from 5 to 1
replace $M1 = 5$ by $M1 = 1$
- Change of output tape logical designation from 6 to 1
replace $M2 = 6$ by $M2 = 1$
- Change of maximum number of lines per printout page from 43 to k
replace $LINEX = 43$ by $LINEX = k$
- 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 $IBMDCD = 0$ by $IBMDCD = 1$

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:

$R = \text{RANF}(i)$, where $i > 0$, stores i as the generator
 $R = \text{RANF}(0)$ 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.

FORTAN IV LISTING IS PRESENTED ON PAGES 184 THROUGH 259.

```

SIMTFC FASTER M94/2,XR7 FAST0001
CFASTFC=FASTFC-ANALYTIC SOLUTION-TRANSPOSE, EQUATION-RANDOM SAMPLING+TH,FAST0002
COMMON N(11200) FAST0003
COMMON/DCID/18MDCD FAST0004
COMMON/TAPEID/M1,M2 FAST0005
COMMON/CASEID/CASE,NPAGE,LINES,LINEX,TITLEA(18),TITLEB(18)FAST0006
COMMON/LIMITS/VSOTF,NERPDR,NSMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0007
1 NEMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0008
2 NUNITX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0009
3 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0010
4 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0011
5 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0012
6 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0013
COMMON/INPUTS/INTP,IAZ,ITSV,INIL,IAH,IAH,IAH,IAH,IAH,FAST0014
1 IFLW,IAE,IBE,INSG,INPC,INJSN,INJSN,FAST0015
2 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0016
3 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0017
4 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0018
5 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0019
6 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0020
7 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0021
8 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0022
9 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0023
10 ITP,ITPV,ITATN,ITATN,ITATN,ITATN,ITATN,ITATN,FAST0024
COMMON/OTHERS/MATJUS,XCT(13),DELTA,BDC(13),ATA,ATF,ATC,FAST0025
1 ATD,AT,AT,AT,AT,AT,AT,AT,AT,AT,AT,AT,AT,AT,AT,AT,FAST0026
2 KNIN,KPAX,JRAR,NZERO,NZERO,NZERO,NZERO,NZERO,NZERO,FAST0027
COMMON/REGSCA/NSRPA,NSR,FAST0028
18MDCD = 0 FAST0029
NSTORE = 12001 FAST0030
M1 = 5 FAST0031
M2 = 6 FAST0032
LINEX = 43 FAST0033
KASF = 0 FAST0034
1) CALL DEFINT FAST0035
CALL SOLVIT FAST0036
GO TO 10 FAST0037
END FAST0038
SIMTFC LABEL M94/2,XR7 FAST0039
CLABEL OUTPUT PAGE HEADINGS FOR PROGRAM FASTER+T.M.JORDAN+MAM+OCT.1966+FAST0040
FUNCTION LABEL(LIKE) FAST0041
COMMON/TAPEID/M1,M2 FAST0042
COMMON/CASEID/CASE,NPAGE,LINES,LINEX,TITLEA(18),TITLEB(18)FAST0043
L = TAUSLINE FAST0044
IF(LINE-LE-OSD TC 10) FAST0045
LINES = LINES + L FAST0046
LABEL = 0 FAST0047

```

```

IF(LINES-LE-LINEKIGD TO 20 FAST0048
1) LINES = L + 3 FAST0049
NPGT = NPGT + 1 FAST0050
LABEL = 1 FAST0051
WRITE(142,2000)TITLEA,CASE,TITLEB,NPAGE FAST0052
20) JUMPT(181,1844,JOH****THE FASTER CODE*****CASE,15/1X,1844,JOH****FAST0053
1) T.M.JORDAN,MAM****PAGE,15/1X 1) FAST0054
2) RETURN FAST0055
END FAST0056
SIMTFC VECTOR M94/2,XR7 FAST0057
CVECTOR UNIT VECTOR FAST0058
FUNCTION VECTOR(X,Y,C) FAST0059
DIMENSION X(1),Y(1),C(1) FAST0060
CST = 0.0 FAST0061
DO 10 I=1,3 FAST0062
C(I) = Y(I) - X(I) FAST0063
1) CST = CST + C(I)**2 FAST0064
CST = SQRT(CST) FAST0065
IF(CST-GE-0.01) GO TO 15 FAST0066
C(1) = 0.0 FAST0067
C(2) = 0.0 FAST0068
C(3) = 1.0 FAST0069
GO TO 25 FAST0070
15) DO 20 I=1,3 FAST0071
2) C(I) = C(I)/CST FAST0072
25) VECTOR = CST FAST0073
RETURN FAST0074
END FAST0075
SIMTFC BLANK M94/2,XR7 FAST0076
CJERUIT=ZERO OUT ARRAY FAST0077
SUBROUTINE JERUIT(MAX,N) FAST0078
DIMENSION N(1) FAST0079
IF(MAX-LE-CLIGD TO 20 FAST0080
DO 1) I=1,MAX FAST0081
1) N(I) = 0 FAST0082
2) RETURN FAST0083
END FAST0084
SIMTFC LOCATF M94/2,XR7 FAST0085
CLOCATF=LOCATF INDEF CALCULATICH FJR PROGRAM FASTER+T.M.JORDAN+MAM+OCT.1966+FAST0086
FUNCTION LOCATF(INN,XXX) FAST0087
DIMENSION XXX(1) FAST0088
COMMON N(1) FAST0089
COMMON/LIMITS/VSOTF,NERPDR,NSMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0090
1 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0091
2 NUNITX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0092
3 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0093
4 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,FAST0094

```

```

5      NVOPAX,NVOPDC,NPCINT,MODELQ,MODELU,MODELV, FAST0094
6      NPRINT,NUNITS,NUMBER,KALIDE FAST0095
COMMON/INDEXS/INTP ,IAZ ,IISV ,IRTL ,IRMO ,IXR ,ITEL , FAST0096
1      ICLB ,IAF ,IBE ,INSG ,INPC ,IJSN ,IJSK , FAST0097
2      IATR ,ISUV ,IATN ,IATW ,IFSO ,ISSH ,IDEN , FAST0098
3      INTC ,ITEL ,IFGW ,ITOS ,IDIV ,IDIR ,IDOS , FAST0099
4      IVOL ,ICDT ,ICDT ,IRSI ,IALP ,IVMD ,IGIM , FAST0100
5      IAIM ,IALM ,IALH ,IA ,INS ,IVCE ,IVAL , FAST0101
6      ISPA ,ISPE ,IATO ,IRSP ,ISGT ,IIDE ,IDDI , FAST0102
7      ISGS ,NEXI ,IXMP ,ISGR ,IWS ,IFS ,INC , FAST0103
8      ITC ,IND ,IU ,IV ,IW ,ISI ,IST , FAST0104
9      INRG ,INSC ,ISTP ,INRP ,INCP ,IFXP ,IFXS , FAST0105
1     IFXT ,IFXE ,IFXA , FAST0106
COMMON/DOUMLC/NN ,NN ,X(19) FAST0107
DO 10 I=1,3 FAST0108
1) X(I) = XXX(I) FAST0109
NN = NN FAST0110
CALL LOCUDM(NMAX,NMAX+1,H(INTP),H(IAZ),H(IAI),H(INS),H(IND), FAST0111
1     H(IU)) FAST0112
LOCATE = MH FAST0113
RETURN FAST0114
END FAST0115
$IFTC LOCDEL M94/2,XR7 FAST0116
LOCUDM=REGIIN INDEX CALCULATION FOR PROGRAM FASTER*T.M.JORDAN*WAL*1966 FAST0117
SUBROUTINE LOCUDM(L1,L2,L3,INTP,IAZ,INS,NDU) FAST0118
DIMENSION NTP(1),AZ(1),AIL(1,3),NS(1,2,1,3),ND(1),U(1) FAST0119
COMMON/DOUMLC/NN ,NN ,X(19) FAST0120
COMMON/LIMITS/NSICRE,NERROR,NSMAX ,NMAX ,NRMAX ,NMAX ,NSMAX, FAST0121
1     NEMAX ,NPMAX ,NPMAX ,NEMAX,NEMOD ,NXCSECT,NUNITD, FAST0122
2     NUNITX,NIMAX ,NPMAX ,NCRDER,NODOWN ,INELAS,NTRANS, FAST0123
3     NRMAX ,NMAX ,NPMAX ,NVMD ,NMAX ,NLMAX ,NMAX , FAST0124
4     NDMAX ,NMOD ,NMDENT,NOMOD ,NPMAX ,NPMOD ,NSDMAX , FAST0125
5     NVDMAX,NVOPDC,NPCINT,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)GL TO 130 FAST0135
IMIN = NN FAST0136
IF(IMIN-1)*NMAX-(MIN+1).GT.C)GO TO 120 FAST0137
IRIX = NRMAX + 1 FAST0138
GO TO 200 FAST0139
12) 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,NRMAX FAST0145
IF(NS(J,1).EQ.0)GO TO 180 FAST0146
KP = NS(J,1)/1000000 FAST0147
K = NS(J,1)/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 = 1 FAST0157
GO TO 210 FAST0158
190 CONTINUE FAST0159
200 CONTINUE FAST0160
NN = 0 FAST0161
210 RETURN FAST0162
END FAST0163
$ORIGIN ALPHA FAST0164
$IFTC DEFIN M94/2,XR7 FAST0165
DEFINE=DATA INPUT AND PREPARATION FOR PROGRAM FASTER*T.M.JORDAN*WAL*66 FAST0166
SUBROUTINE DEFINE FAST0167
COMMON FAST0168
COMMON/TAPEID/MI ,M2 FAST0169
COMMON/CASEID/KASE ,NPAGE ,LINES ,LINEX ,TITLEA(18) ,TITLEB(18) FAST0170
COMMON/LIMITS/NSSTORE,NERROR,NSMAX ,NMAX ,NRMAX ,NMAX ,NSMAX, FAST0171
1     NEMAX ,NPMAX ,NPMAX ,NEMAX,NEMOD ,NXCSECT,NUNITD, FAST0172
2     NUNITX,NIMAX ,NPMAX ,NCRDER,NODOWN ,INELAS,NTRANS, FAST0173
3     NRMAX ,NMAX ,NPMAX ,NVMD ,NMAX ,NLMAX ,NMAX , FAST0174
4     NDMAX ,NMOD ,NMDENT,NOMOD ,NPMAX ,NPMOD ,NSDMAX , FAST0175
5     NVDMAX,NVOPDC,NPCINT,MODELQ,MODELU,MODELV, FAST0176
6     NPRINT,NUNITS,NUMBER,KALIDE FAST0177
COMMON/INDEXS/INTP ,IAZ ,IISV ,IRTL ,IRMO ,IXR ,ITEL , IFLL , FAST0178
1     ICLB ,IAE ,IBE ,INSG ,INPC ,IJSN ,IJSK , FAST0179
2     IATR ,ISUV ,IATN ,IATW ,IFSO ,ISSH ,IDEN , FAST0180
3     INTC ,ITEL ,IFGW ,ITOS ,IDIV ,IDIR ,IDOS , FAST0181
4     IVOL ,ICDT ,ICDT ,IRSI ,IALP ,IVMD ,IGIM , FAST0182
5     IAIM ,IALM ,IALH ,IA ,INS ,IVCE ,IVAL , FAST0183
6     ISPA ,ISPE ,IATO ,IRSP ,ISGT ,IIDE ,IDDI , FAST0184
7     ISGS ,NEXI ,IXMP ,ISGR ,IWS ,IFS ,INC , FAST0185
8     ITC ,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(73)          FAST0189
DIMENS ION TOUM(36),TYPE I5,6)      ,ADUM(18) FAST0190
DATA TOUM,                            FAST0191
1 4HIFID,4HRTA,4HN,14,4H1NAL,4HYTIC,4H (S1,4HOLUT,4HION,4HITR),FAST0192
2 4HASP,4HRT,4HEIO,4HUAT1,4HON,1,4HRIAN,4HDOM,4HSAMP,4HLING,FAST0193
3 4HMAF,4HEMAT,4HICAL,4H AND,4H NUN,4HERIC,4HAL,4HNALY,4HSIS,FAST0194
4 4HCNP,4HUTER,4H PRG,4HGRAM,4HMG,4H AND,4H CHE,4HCKD,4HT BY,FAST0195
DATA TYPE/,                            FAST0196
1 4H VAR,4HIAB,4HE D1,4HMENS,4HIONS, FAST0197
2 4HSURF,4HACES,4H ANC,4H REG,4HIGAS, FAST0198
3 4H IQ,4HEPEA,4HIDENT,4H SOU,4HRCES, FAST0199
4 4HBA51,4HCE CB,4HMES,4HSECT,4HIONS, FAST0200
5 4HDETE,4HCTOR,4HS +,4HRESP,4HUNSE, FAST0201
6 4H SAM,4HPLI,4HG PA,4HNAME,4HTERS, FAST0202
10 KASE = KASE + 1                     FAST0203
NPAGE = 0                             FAST0204
LINES = LINEX                          FAST0205
IF(KASE.GT.1) GO TO 3C                 FAST0206
DO 20 J=1,36                          FAST0207
70 TITLE(J) = TOUM(J)                 FAST0208
70 CONTINUE                           FAST0209
DO 700 NNN=1,6                        FAST0210
IF(LABEL(1).GE.0) WRITE(2,2000) NNN,TYPE(1,NNN),I=1,5) FAST0211
2000 FORMAT(1X,30(1H),18NDATA INPUT SECTION,12,2H, ,5A,33(1H*)) FAST0212
CALL READI(24,IN1)                   FAST0213
IF(IN1.LE.01) GO TO 6C                 FAST0214
DO 50 I=1,IN1                         FAST0215
50 CALL READA(18,ADUM)                FAST0216
60 GO TO (100,200,30C,4C0,500,600), NNN FAST0217
100 CALL STORER                       FAST0218
GO TO 700                             FAST0219
200 CONTINUE                         FAST0220
CALL GEDIN(INMAX,NMAX,1,V1INTP,V1IAZ),V1IA),V1ISV),V1IMT), FAST0221
1 V1INS),V1IRHO),V1IXR1)              FAST0222
GO TO 700                             FAST0223
300 IE = NSTORE - NXEMAX               FAST0224
IFN = IE - NXEMAX                     FAST0225
IFNG = IEN - NXEMAX                   FAST0226
IERO = IENG - NXEMAX                  FAST0227
IF(IEBG.LT.NEXT) NERROR = NERROR + 1 FAST0228
CALL SOURCE(INMAX,NMAX,5,1,V1IELL),V1IELW),V1IAE),V1IEF),V1ISGT),FAST0229
1 V1INCP),V1IJSK),V1IJSU),V1ISUV),V1IXTR),V1IVEF), FAST0230
2 V1IAL),V1ISPA),V1ISPE),V1ISV),V1IE 1,V1IEN), FAST0231
3 V1IEBG),V1IFNG)                     FAST0232
GO TO 700                             FAST0233
400 IEAC = NSTOPE - NEMOD*(INMAX + NMAX) FAST0234

```

```

1JMX = IEAC - NXSECT*(INELAS + 3)     FAST0235
IXSI = IJMX - NXSECT*INELAS*NTTRANS FAST0236
IXSE = IXSI - NXSECT* NDOMX*NEMAX*MAXO(2,NORDER) FAST0237
IXST = IXSE - NEMOD                   FAST0238
INI = IXST - ISGS                     FAST0239
CALL INSECT(NEMOD,NDOWN,NORDER,INELAS,NEMAX,1,V1IESB),V1IDEN), FAST0240
1 V1ISSH),V1ISGT),V1IDEL),V1ID1) ,V1ISES), FAST0241
2 V1IMTL),V1IRHO),V1IXST),V1IATD),V1IJMK),V1IXSE), FAST0242
3V1IXST),V1IEAC),V1IELL),NTRANS)     FAST0243
NEXT = ISGS + NMAX                    FAST0244
IF(NEXT.GT.IXMP) NERROR = NERROR + 1 FAST0245
GO TO 700                             FAST0246
500 CALL RFSULT(INGMOD,1,V1INTG),V1IELF),V1IFGW),V1ITOS),V1IRSP), FAST0247
1 V1ITOR),V1IIDS),V1IVOL),V1ICDT),V1IXDT),V1IELL), FAST0248
2V1IMTL),V1IRHO),V1IEAC),NEMOD)      FAST0249
GO TO 700                             FAST0250
600 CALL RANDM(V1IRSI),V1IALP),V1IVMD),V1IGIM),V1IAMI),V1IALM), FAST0251
1V1IALH),V1INCP),V1IVEE),NMAX,5)     FAST0252
200 CONTINUE                         FAST0253
IF(LABEL(1).GE.0)WRITE(2,2010)        FAST0254
2010 FORMAT(1X,35(1H*),36NDATA INPUT AND PREPARATION CCMPLETED,36(1H*)) FAST0255
IF(NERROR.EC.0) GO TO 900             FAST0256
IF(LABEL(1).GE.0)WRITE(2,2020)        FAST0257
2020 FORMAT(1X,42(1H*),23HPROCEEDING TO NEXT CASE,42(1H*)) FAST0258
GO TO 10                              FAST0259
900 MAX = NSTORE - 1                  FAST0260
DO 910 I=NEXT,MAX                     FAST0261
910 V(I) = 0.0                        FAST0262
IF(LABEL(1).GE.0)WRITE(2,2040)        FAST0263
2040 FORMAT(1X,1H*,35(3HGD*))         FAST0264
LINES = LINEX + 1                     FAST0265
RETURN                                FAST0266
END                                    FAST0267
18BFC STORE N94/2,XR7                 FAST0268
CSTORER=LIMIT INPUT AND VARIABLE DIMENSIONING FOR PROGRAM FASTER*TKJ*66* FAST0269
SUBROUTINE STORER                     FAST0270
COMMON N(1)                          FAST0271
COMMON/REGSCA/NSRMAX,INSR           FAST0272
COMMON/TAPED/MI ,M2                  FAST0273
COMMON/CASEID/KASE ,NPAGE ,LINES ,LINEX ,TITLEA(18) ,TITLEB(18) FAST0274
COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NMAX ,NRMAX ,NOMAX ,NSTMAX, FAST0275
1 NMAX ,NEMAX ,NMAX ,NXEMAX,NEMOD ,NXSECT,UNITD, FAST0276
2 NUNITX,NINAX ,NNMAX ,NORDER,NODWN ,INELAS,NTRANS, FAST0277
3 NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCHAX ,NLMAX ,RTMAX , FAST0278
4 NOMAX ,NGMOD ,HOPENT ,NOMOD ,NPDMAX ,NPOMOD ,NSDMAX, FAST0279
5 NVDMAX,NVDMOD,NPOINT,NODELP,NODELV,NODELV, FAST0280
6 NPRINT,NUNITS,NUMBER,KALIDE         FAST0281

```

```

COMMON/INDEXS/INVP      ,IAZ      ,IISV      ,IMTL      ,IRHO      ,IXR      ,IELL      , FASTO283
1      IELM      ,IAE      ,IBE      ,INSG      ,INPC      ,IJSN      ,IJSX      , FASTO284
2      IXTN      ,ISUV      ,IATN      ,IATM      ,IESB      ,ISSH      ,IDEN      , FASTO285
3      INTG      ,IELF      ,IFOW      ,ITDS      ,ITDV      ,ITDR      ,ITDS      , FASTO286
4      IVOL      ,ICDT      ,IXDT      ,IRSI      ,IALP      ,IVMO      ,IGIM      , FASTO287
5      IATH      ,IALM      ,IALH      ,IA      ,INS      ,IVEE      ,IVAL      , FASTO288
6      ISPH      ,ISPE      ,IATD      ,IRSP      ,ISGT      ,IIDE      ,ITDI      , FASTO289
7      ISGS      ,NEXT      ,IXMP      ,ISGR      ,INS      ,IES      ,IWC      , FASTO290
8      IEC      ,IND      ,IU      ,IV      ,IW      ,ISI      ,IST      , FASTO291
9      INRG      ,INSC      ,ISTP      ,INRP      ,INCP      ,IFXP      ,IFXS      , FASTO292
1     IFXT      ,IFAE      ,IFXA
COMMON/INPUTS/IN1      ,IN2      ,IN3      ,IN4      ,IN5      ,IN6      ,IN7      , FASTO293
1     IN8      ,IN9      ,IN10      ,IN11      ,IN12      ,IN13
DIMENSION LCP(51),ARRAY(73),LOC(1)
EQUIVALENCE (LOC(1),INPP)
EQUIVALENCE(LCP(13),L1),LCP(40),L2),LCP(41),L3),LCP(42),L4),
LCP(43),L5),LCP(44),L6),LCP(45),L7),LCP(46),L8),LCP(47),L9),
LCP(48),L10),LCP(49),L11),LCP(50),L12)
DATA ARRAY / 3HNTF,3HAE,3HVS,3HMTL,3HRHO,3HXR,3HELL,FASTO298
13HELM,3HAE,3HBE,3HNSG,3HNPC,3HJSN,3HJSX,3HXTA,3HSUV,3HATN,3HATH,FASTO299
13HESB,3HSSH,3HDEN,3HNTG,3HELF,3HFGM,3HTDS,3HIDV,3HIDR,3HIDS,3HVOL,FASTO300
13HCDT,3HXDT,3HRSI,3HALP,3HWD,3HGM,3HALM,3HALM,3HA,3HNS,FASTO301
13HVEE,3HVAL,3HSPH,3HSP,3HATD,3HRS,3HGT,3HIDE,3HIDI,3HSGS,3HEND,FASTO302
13HXMP,3HSGR,3HNS,3HES,3HMC,3HEC,3HND,3HU,3HV,3HSI,FASTO303
13HST,3HNRG,3HNSC,3HSTP,3HNR,3HNCP,3HXP,3HFX,3HXT,3HFXE,3HFXA,FASTO304
13HASE,GT,1) GO TO 20
NEORR = 0
NNMAX = 0
MAX = NSTORE - 1
DO 10 I=1,NAX
10 N(I) = 0
GO TO 30
20 MAXS = NSMAX
MAXA = NMAX
MAXR = NRMAX
MAXB = NBMAX
MAXV = NVMAX
MAXW = NWMAX
MAXE = NEMAX
MAXM = NMMAX
MAXI = NIMAX
MAXF = NFMAX
MODG = NGMOD
MODE = NEMOD
MODD = NXSECT
MODD = NORDER
MODD = NODDN
FASTO295
FASTO296
FASTO297
FASTO298
FASTO299
FASTO300
FASTO301
FASTO302
FASTO303
FASTO304
FASTO305
FASTO306
FASTO307
FASTO308
FASTO309
FASTO310
FASTO311
FASTO312
FASTO313
FASTO314
FASTO315
FASTO316
FASTO317
FASTO318
FASTO319
FASTO320
FASTO321
FASTO322
FASTO323
FASTO324
FASTO325
FASTO326
FASTO327
FASTO328

```

```

MODI = INELAS
MAXM = NMMAX
30 IF(IN2,GT,0) CALL READA(18,TITLEA)
IF(IN3,GT,0) CALL READA(18,TITLEB)
IF(IN4,GT,0) CALL READI(5,NSMAX)
IF(IN5,GT,0) CALL READI(4,NEMAX)
IF(IN6,GT,0) CALL READI(5,NXSECT)
IF(IN7,GT,0) CALL READI(4,NORDER)
IF(IN8,GT,0) CALL READI(8,NCPAX)
IF(IN9,GT,0) NSRMAX = NGMOD
IF(IN9,GT,0) CALL READI(5,NPCINT)
IF(IN10,GT,0) CALL READI(4,NPRINT)
IF(NASE,EQ,1) GO TO 70
NARRAY = 51
IF(NXSECT,EQ,NMODX) GO TO 40
NARRAY = 47
NNMAX = 0
40 MOVE = LOC(NARRAY)
NMOVE = NSTORE - MOVE
DO 50 I=1,NARRAY
50 LCP(I) = LOC(I) + NMOVE
I = NSTORE
J = MOVE
NMOME = MOVE - 1
DO 60 K=1,NMOVE
I = I - 1
J = J + 1
60 NMOM = NJU
70 NMOM = NEMAX + 1
NGMOD = NGMAX + 1
NMOMENT = NVMOD + NCMAX + NLMAX + NTMAX + 1
NMOMENT = NMOMENT + NSRMAX
NMOM = 1
IF(NPOINT,EQ,0) NMOM = NMOM
INTP = NSRMAX + 1
IAZ = INTP + NSMAX
IISV = IAZ + NSMAX
IMTL = IISV + NRMAX
IRHO = IMTL + NRMAX
IXR = IRHO + NRMAX
IELL = IXR + NRMAX*3
IELM = IELL + NEMOD
IAE = IELM + NEMAX
IBE = IAE + NEMAX
INSG = IBE + NEMAX
INPC = INSG + NVMAX
IJSN = INPC + NVMAX*5
FASTO329
FASTO330
FASTO331
FASTO332
FASTO333
FASTO334
FASTO335
FASTO336
FASTO337
FASTO338
FASTO339
FASTO340
FASTO341
FASTO342
FASTO343
FASTO344
FASTO345
FASTO346
FASTO347
FASTO348
FASTO349
FASTO350
FASTO351
FASTO352
FASTO353
FASTO354
FASTO355
FASTO356
FASTO357
FASTO358
FASTO359
FASTO360
FASTO361
FASTO362
FASTO363
FASTO364
FASTO365
FASTO366
FASTO367
FASTO368
FASTO369
FASTO370
FASTO371
FASTO372
FASTO373
FASTO374
FASTO375
FASTO376

```

```

IJSX = IJSN + NVMAX
IXTR = IJSX + NVMAX
ISUV = IXTR + NVMAX*3
IATN = ISUV + NVMAX
IATW = IATN + NIMAX
IESB = IATW + NIMAX
ISSH = IESB + NEMAX
IDEN = ISSH + NEMOD
INTG = IDEN + NMKAX
IELF = INTG + NEMAX
IFGW = IELF + NGMOD
ITDS = IFGW + NDMAX
ITOV = ITGS + NFMAX*3
IIDR = ITOV + NVMOD
IDOS = IIDR + NDMAX
IVOL = IDOS + NDMAX
ICOT = IVOL + NDMAX
IXOT = ICOT + NDMAX*3
IRSI = IXOT + NDMAX*3
IALP = IRSI + NVMAX
IVMO = IALP + NVMAX*5
IGIM = IVMO + NVMAX*5
IAIM = IGIM + NEMAX
IALM = IAIM + NEMAX
IALH = IALM + NEMAX
IA = IALH + NEMAX
INS = IA + NSMAX*NMKAX
IVEE = INS + NMKAX*NDMAX
IVAL = IVEE + NVMAX*NXMAX*5
ISPW = IVAL + NVMAX*NXMAX*5
ISPE = ISPW + NVMAX*NEMAX
IATO = ISPE + NVMAX*NEMAX
IRSP = IATO + NMKAX*INMAX
ISGT = IRSP + NFMAX*NGMOD
IDOE = ISGT + NMKAX*NEMOD
IIDO = IDOE + NMKAX*NGMOD*NDOWN*NXSECT
ISGS = IIDO + NMKAX*INELAS*NXSECT
NEXT = ISGS + NMKAX*NXSECT
IFXA = NSTORE - NDOD*NGMAX*MODRENT
IFXE = IFXA - NDOD*NGMAX
IFXT = IFXE - NDOD*NGMAX
IFXS = IFXT - NDOD*NGMAX
IFXP = IFXS - NDOD*NGMAX
INCP = IFXP - NSTMAX
INRP = INCP - NSTMAX
INSG = ISTP - NSTMAX

```

FAST0377
FAST0378
FAST0379
FAST0380
FAST0381
FAST0382
FAST0383
FAST0384
FAST0385
FAST0386
FAST0387
FAST0388
FAST0389
FAST0390
FAST0391
FAST0392
FAST0393
FAST0394
FAST0395
FAST0396
FAST0397
FAST0398
FAST0399
FAST0400
FAST0401
FAST0402
FAST0403
FAST0404
FAST0405
FAST0406
FAST0407
FAST0408
FAST0409
FAST0410
FAST0411
FAST0412
FAST0413
FAST0414
FAST0415
FAST0416
FAST0417
FAST0418
FAST0419
FAST0420
FAST0421
FAST0422

```

INRG = INSC - NSTMAX
IST = INRG - NSTMAX
ISI = IST - NSMAX*2
IW = IST - NSMAX
IV = IW - NSMAX
IU = IV - NSMAX
IND = IU - NSMAX
IEC = IND - NEMAX
IWC = IEC - NEMAX
IES = IWC - NEMAX
IWS = IES - NEMAX
ISGR = IWS - NEMAX
IXMP = ISGR - NEMAX
DO 90 I=1,38
MOD = LCP(I+1) - LCP(I)
MAX = LOC(I+1) - LOC(I)
IF(MOD*MAX).EQ.0) GO TO 90
IF(LOC(I).GT.LCP(I)) GO TO 80
IF(LOC(I+1).GT.LCP(I+1)) GO TO 80
J = LOC(I)
K = LCP(I)
CALL RESTOW(MAX,1,1,N(I),MOD,1,1,N(K))
GO TO 90
80 IF(LABEL(1).GE.0) WRITE(M2,2000)ARRAY(I)
2000 FORMAT(1X,39(1H*),A3,25H ARRAY CANNOT BE RESTORED,40(1H*))
NERROR = NERROR + 1
90 CONTINUE
DO 100 I=40,NARRAY
IF(LCP(I-1).GE.LOC(I)) GO TO 100
IF(LABEL(1).GE.0) WRITE(M2,2010)ARRAY(I-1)
2010 FORMAT(1X,35(1H*),A3,34H ARRAY MAY BE RESTORED INCORRECTLY,35(1H*))
NERROR = NERROR + 1
100 CONTINUE
CALL RESTCWIN(MAX,NSMAX,1,N(I),MAX,MAX,1,N(L1))
CALL RESTWIN(MAX,NRMAX,1,N(INS),MAX,MAX,1,N(L2))
CALL RESTWIN(MAX,5,NVMAX,N(IVEE),MAX,5,MAX,N(L3))
CALL RESTWIN(MAX,5,NVMAX,N(IVEE),MAX,5,MAX,N(L4))
CALL RESTWIN(MAX,NVMAX,1,N(ISPW),MAX,MAX,1,N(L5))
CALL RESTWIN(MAX,NVMAX,1,N(ISPE),MAX,MAX,1,N(L6))
CALL RESTCWIN(MAX,NHMAX,1,N(IATD),MAX,MAX,1,N(L7))
CALL RESTWIN(MOD,NFMAX,1,N(IRSP),MOD,MAX,1,N(L8))
IF(NXSECT.NE.MODC) GO TO 120
CALL RESTWIN(MOD,NFMAX,1,N(ISC1),MOD,MAX,1,N(L9))
IF(NXSECT.EC.0) GO TO 120
CALL RESTWIN(MOD,NMOD,NFMAX,N(IJDE1),MOD,MOD,MAX,N(L10))

```

FAST0423
FAST0424
FAST0425
FAST0426
FAST0427
FAST0428
FAST0429
FAST0430
FAST0431
FAST0432
FAST0433
FAST0434
FAST0435
FAST0436
FAST0437
FAST0438
FAST0439
FAST0440
FAST0441
FAST0442
FAST0443
FAST0444
FAST0445
FAST0446
FAST0447
FAST0448
FAST0449
FAST0450
FAST0451
FAST0452
FAST0453
FAST0454
FAST0455
FAST0456
FAST0457
FAST0458
FAST0459
FAST0460
FAST0461
FAST0462
FAST0463
FAST0464
FAST0465
FAST0466
FAST0467
FAST0468
FAST0469

```

      IF(INELAS.LE.0) GO TO 110
      CALL RESTOM(INELAS,NPMAX,1,N1(101),MODI,MAXN,1,N1(111))
110 CALL RESTOM(NMAX,1,1,N1(55),MAXN,1,1,N1(21))
120 IF(IN11.LE.0) GO TO 130
      IF(LABEL(15).GE.0) WRITE(12,2030) (ARRAY(I), LGC(I), I=1,73)
130 CONTINUE
2030 FORMAT(1X,43(1H*,1,2)NDATA ARRAY STCRAGE MAP,43(1H*,1/6(4X,14HARRAY
      ILOCATION)/16(6X,A3,19)))
      RETURN
      END
S18FTC RESTOR M94/2,XR7
CRSTOM*RESTORE MULTIDIMENSIONAL ARRAYS WITH NEW DIMENSIONS*F. M. JORDAN*WANH*66
      SUBROUTINE RESTOM(I,J,K,K1,NB,J1,NB,NB)
      DIMENSION NA(1:A,J,K),NB(1:B,J,K)
      DO 20 K=1,K1
      DO 20 J=1,J1
      DO 10 I=1,IA
      NC = 0
      IF(I.GT.18) GO TO 10
      IF(J.GT.38) GO TO 10
      IF(K.GT.38) GO TO 10
      NC = NB(I,J,K)
10 NA(I,J,K) = NC
20 CONTINUE
      RETURN
      END
S18FTC GEOM M94/2,XR7
CGEOMIN*SURFACE AND REGION INPUT FOR PROGRAM FASTER*F. M. JORDAN*WANH*66
      SUBROUTINE GEOMIN(L,L2,L3,NTP,AZ,A,ISV,MTL,NS,RHO,XR)
      DIMENSION AA(7),ADM(9),IDM(12)
      EQUIVALENCE (AA(1),ADM(1))
      DIMENSION NTP(1),AZ(1),A(1,L3),ISV(1),MTL(1),NS(L2,L3),RHO(1),
      1 XR(3,1)
      COMMON/TAPEIO/M1,M2
      COMMON/LIMITS/NSSTORE,NERROR,NSMAX,NMAX,NMAX,NMAX,NMAX,NMAX,
      1 NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,NMAX,
      2 NUNITX,NMAX,NMAX,NORDER,NDOON,INELAS,NTRANS,
      3 NMAX,NMAX,NMAX,NVMOO,NCMAX,NLMAX,NTMAX,
      4 NMAX,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,
      5 NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,NVMOO,
      6 NPRINT,NUNITX,NUMBER,KALIDE
      COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)
C QUADRIC SURFACES
      IF(IN2.LE.0) GO TO 300
      DO 290 N=1,IN2
      CALL READS(ION,AA)
      I = IDM(1)

```

FAST0470
 FAST0471
 FAST0472
 FAST0473
 FAST0474
 FAST0475
 FAST0476
 FAST0477
 FAST0478
 FAST0479
 FAST0480
 FAST0481
 FAST0482
 FAST0483
 FAST0484
 FAST0485
 FAST0486
 FAST0487
 FAST0488
 FAST0489
 FAST0490
 FAST0491
 FAST0492
 FAST0493
 FAST0494
 FAST0495
 FAST0496
 FAST0497
 FAST0498
 FAST0499
 FAST0500
 FAST0501
 FAST0502
 FAST0503
 FAST0504
 FAST0505
 FAST0506
 FAST0507
 FAST0508
 FAST0509
 FAST0510
 FAST0511
 FAST0512
 FAST0513
 FAST0514
 FAST0515
 FAST0516

```

      NTP(I) = 10*(2)
      NEX = 10*(3)
      IF(NEX.GT.0) GO TO 230
C EXPANDED FORM
      AZ(I) = AA(1)
      HAX = MIN(INMAX,6)
      DO 220 J=1,HAX
220 A(J,I) = AA(J+1)
      IF(NTP(I).GT.6) CALL READS(NTP(I)-6,A(7,1))
      GO TO 290
230 NGT = (NEX+2)/3
      NEX = NEX - 3*(NGT-1)
      DO 231 J=1,NMAX
231 A(J,I) = 0.0
      GO TO(232,233,235,260,255),NGT
232 NTP(I) = NEX
      AZ(1) = -AA(1)
      A(NEX,1) = 1.0
      GO TO 290
233 FST = AA(3) - AA(1)
      D = -1.0
      K = 0
      DO 234 J=1,3
      IF(J.EQ.NEX) GO TO 234
      D = 0
      K = K + 1
      NTP(I) = J
      A(J,I) = D*FST/(AA(K+2) - AA(K))
      AZ(1) = AZ(1) - AA(K)*A(J,I)
234 CONTINUE
      GO TO 290
C CDNE PARALLEL TO NEX-AXIS
235 CONTINUE
      XX = (AA(6) - AA(4))/(AA(5) - AA(3))
      YY = XX*AA(3) - AA(4)
      NTP(I) = 6
      XX = XX**2
      AZ(1) = YY**2 - XX*(AA(1)**2 + AA(2)**2)
      K = 0
      DO 250 J=1,3
      IF(J.EQ.NEX) GO TO 240
      A(J,I) = 2.0*YY
      A(J+3,I) = 1.0
      GO TO 250
240 K = K + 1
      A(J,I) = 2.0*XX*AA(K)

```

FAST0517
 FAST0518
 FAST0519
 FAST0520
 FAST0521
 FAST0522
 FAST0523
 FAST0524
 FAST0525
 FAST0526
 FAST0527
 FAST0528
 FAST0529
 FAST0530
 FAST0531
 FAST0532
 FAST0533
 FAST0534
 FAST0535
 FAST0536
 FAST0537
 FAST0538
 FAST0539
 FAST0540
 FAST0541
 FAST0542
 FAST0543
 FAST0544
 FAST0545
 FAST0546
 FAST0547
 FAST0548
 FAST0549
 FAST0550
 FAST0551
 FAST0552
 FAST0553
 FAST0554
 FAST0555
 FAST0556
 FAST0557
 FAST0558
 FAST0559
 FAST0560
 FAST0561
 FAST0562
 FAST0563

Astronuclear
Laboratory

Astronuclear
Laboratory

```

10 RE(I) = 0.0
C TOTAL SOURCE, ERRGR CRITERION, TRANSLATION VECTOR
330 IF(I*3,LE,0)GO TO 505
DO 430 NNN=1,IN3
CALL READR(10M,ADP)
VOLUME = 1.0
I = (DM(I))
NSG(I) = (DM(2))
N = NSG(I)
DO 405 J=1,5
MAX = (DM(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)405,405,21C
299 NPC(J,I) = MAX
CALL READEE(MAX,VEE(I,J,I),VAL(I,J,I))
IF(MAX,LE,1)GO TO 405
FST = 0.0
C DO 907 K=1,MAX
TF(VEE(K,J,I),EQ,0,0) VEE(K,J,I) = 1.0E-30
C 907 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)
I = 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 = (DM(10))
MAX = (DM(8))
IF(MAX,GT,0)GO TO 29F
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(10M(11),GT,0) GO TO 600
CALL READEE(MAX,E,EN)
GO TO 601
600 IF(10M(11),EQ,2) CALL READEE(MAX,E)
CALL READEE(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 = NC, GROUPS
100 IF(10M(11),EQ,0) GO TO 351
IF(10M(11),EQ,2) CALL READEE(MAX+1,EBG)
CALL READEE(MAX,ENG)
GO TO 110
351 CALL READEE(MAX+1,EBG,ENG)
110 CONTINUE
IF(ISP,EQ,2)GO TO 355
C TOTAL NEW 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=L+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

```

FAST0658
FAST0659
FAST0660
FAST0661
FAST0662
FAST0663
FAST0664
FAST0665
FAST0666
FAST0667
FAST0668
FAST0669
FAST0670
FAST0671
FAST0672
FAST0673
FAST0674
FAST0675
FAST0676
FAST0677
FAST0678
FAST0679
FAST0680
FAST0681
FAST0682
FAST0683
FAST0684
FAST0685
FAST0686
FAST0687
FAST0688
FAST0689
FAST0690
FAST0691
FAST0692
FAST0693
FAST0694
FAST0695
FAST0696
FAST0697
FAST0698
FAST0699
FAST0700
FAST0701
FAST0702
FAST0703
FAST0704

FAST0705
FAST0706
FAST0707
FAST0708
FAST0709
FAST0710
FAST0711
FAST0712
FAST0713
FAST0714
FAST0715
FAST0716
FAST0717
FAST0718
FAST0719
FAST0720
FAST0721
FAST0722
FAST0723
FAST0724
FAST0725
FAST0726
FAST0727
FAST0728
FAST0729
FAST0730
FAST0731
FAST0732
FAST0733
FAST0734
FAST0735
FAST0736
FAST0737
FAST0738
FAST0739
FAST0740
FAST0741
FAST0742
FAST0743
FAST0744
FAST0745
FAST0746
FAST0747
FAST0748
FAST0749
FAST0750
FAST0751



FAST0799
FAST0800
FAST0801
FAST0802
FAST0803
FAST0804
FAST0805
FAST0806
FAST0807
FAST0808
FAST0809
FAST0810
FAST0811
FAST0812
FAST0813
FAST0814
FAST0815
FAST0816
FAST0817
FAST0818
FAST0819
FAST0820
FAST0821
FAST0822
FAST0823
FAST0824
FAST0825
FAST0826
FAST0827
FAST0828
FAST0829
FAST0830
FAST0831
FAST0832
FAST0833
FAST0834
FAST0835
FAST0836
FAST0837
FAST0838
FAST0839
FAST0840
FAST0842
FAST0843
FAST0844
FAST0845

```

15GS,MTL,RHO,XST,ATO,JMX,XSE,XSI,EAC,ELL,L7)
2  DIMENSION ADM(8)
3  EQUIVALENCE (ADM(1),ATW1,ADM(2),ZEE)
4  DIMENSION ESO(11),DEN(1),SSH(1),SGT(1),L1,DEL(2),L3,L6,I,
5      1 (I(1),L6) ,SS(1:1),MTLE(1),RHQ(1),XST(11),
6      2 ATO(11),JMX(11),XSE(L5,L2,L6),XSI(L7,L6)
7  DIMENSION EAC(L1,L6),ELL(11)
8  COMMON/TAPED/IN1,N2
9  COMMON/LIMITS/MSTORE,NECRDR,NMXAX,NMAX,NRMAX,NBMAX,NSTMXX,
10     COMMON/LIMITS/MSTORE,NECRDR,NMXAX,NXEMAX,NEHOD,NXSECT,NUNITD,
11     1 NEMAX,NVMXX,NXMAX,NMXAX,NXEMAX,NEHOD,NXSECT,NUNITD,
12     NUNITX,NIXMAX,NMXAX,NMRODR,NEDOWN,NIELAS,NTRANS,
13     2 NUNITX,NCRAX,NPEAX,NPEAX,NCRAX,NLMAX,NITAX,
14     3 NMAX,NHOD,NHODM,NHODM,NHODM,NPDMAX,NPDMD,NSDMAX,
15     4 NVDMAX,NVHODM,NPOINT,MODELP,MODELO,MODELU,MODELV,
16     5 NPRINT,NUNITS,NUMBER,KALIDE
17  COMMON/INPUTS/IN1,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,INN(14)
18  DIMENSION XSET(21)
19  IF(IN2,GT,0)CALL READENENAX,ESB)
20  IF(IN3,LE,0)GO TO 210
21  NERRX = 0
22  NEMIX = NEHOD - NXSECT
23  DO 10 I=1,NMXX
24  DEN(I) = 0,0
25  DO 10 J=1,NEMIN
26  SGT(J,I) = 0,0
27  IF(NXSECT,NE,1)GO TO 50
28  DO 30 I=1,NMXX
29  DO 20 J=1,NCRDR
30  DO 20 K=1,NDDNM
31  20 IDE(K,J) = 0
32  CONTINUE
33  IF(INELAS,GT,0)CALL ZEROOUT(INELAS*NMXX,IDO1)
34  CALL ZEROOUT(IN1,SGS)
35  NN = 0
36  NNMOD = IN1 + 1
37  CALL ZEROOUT(NEHOD*(NIMAX+NMXX1),EAC)
38  NN
39  DO 200 I=1,NIMAX
40  MOD = MINO(8,NMXX+2)
41  CALL READENMOD,ADM)
42  MOD = MOD - 2
43  DO 65 J=1,MOD
44  65 ATO(J) = ADM(J+2)
45  IF(NMXX,GT,6)CALL READENNMXX-6,ATO(7))
46  IF(UNFINITO,EG,0)GO TO 85
47  FST = 0.6025/ATW1

```

```

      DO 75 J=1,NMAX
      75 ATD(J) = FST*ATD(J)
      85 CALL READI(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 ABSORPTION COEFFICIENT
      DO 106 J=1,NEMOD
      XMC = ELL(J)/0.511
      XMB = 1.0 + 2.0*XMA
      106 EAC(J,1)=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,1) = 0.0
      IF(J.GT.1) EAC(J,1) = SSH(J)-1
      IF(J.LT.NEMOD) EAC(J,1) = EAC(J,1) + SSH(J)
      116 EAC(J,1) = ELL(J)*EAC(J,1)/4.0
      GO TO 398
      125 IF(NXSECT.EQ.0)GO TO 100
      MOD = MINO(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.2)CALL READI(KMAX-21,JMX(25))
      DO 80 L=1,LMAX
      DO 80 K=1,NDSM
      80 CALL READI(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/C.6025
      LMOD = MINO(2,LMAX)
      DO 82 J=1,NEMAX
      FST = 0.0
      XMB = 0.0
      NDSMOD = MINO(NDSM,NEMAX+1-J)
      DO 81 L=1,LMOD
      DO 81 K=1,NDSMOD
      81 XSE(L) = XSE(L) + XSE(J,K,L)

```

FAST0846
 FAST0847
 FAST0848
 FAST0849
 FAST0850
 FAST0851
 FAST0852
 FAST0853
 FAST0854
 FAST0855
 FAST0856
 FAST0857
 FAST0858
 FAST0859
 FAST0860
 FAST0861
 FAST0862
 FAST0863
 FAST0864
 FAST0865
 FAST0866
 FAST0867
 FAST0868
 FAST0869
 FAST0870
 FAST0871
 FAST0872
 FAST0873
 FAST0874
 FAST0875
 FAST0876
 FAST0877
 FAST0878
 FAST0879
 FAST0880
 FAST0881
 FAST0882
 FAST0883
 FAST0884
 FAST0885
 FAST0886
 FAST0887
 FAST0888
 FAST0889
 FAST0890
 FAST0891
 FAST0892

```

      XMUL = FST/(3.0*XMB)
      XMUC = XMUL**2 - 1.0
      XMUC = (XMUC + XMUL*SQRT(ATW**2 + XMUC))/ATW
      EAC(J,1) = 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 = MINO(NORDER,LMAX)
      XMA = 0.0
      DO 397 J=1,NEMOD
      XMB = 0.0
      IF(J.LT.NEMOD) XMB = EAC(J,1)
      EAC(J,1) = ELL(J)*(XMA + XMB)
      397 XMA = XMB
      IF(KMAX.LE.0)GO TO 398
      C INELASTIC TRANSPORT
      DO 90 K=1,KMAX
      90 CALL READI(JMX(K*3),XSI(1,K))
      398 DO 117 J=1,NEMOD,NEMAX
      117 EAC(J,1) = 2.0*EAC(J,1)
      C COMBINE MATERIAL TOTALS
      100 DO 190 M=1,NMAX
      IF(ATDIM).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,1)
      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
      NM = IDE(K,L,M)
      MAX = NEMOD - K
      IF (NM,GT.0) GO TO 140
      NM = NN
      NN = NM + MAX
      IDE(K,L,M) = NM
      140 IF((NM + MAX).LE.NNMOD) GO TO 150
      NERROR = NERROR + 1

```

FAST0893
 FAST0894
 FAST0895
 FAST0896
 FAST0897
 FAST0898
 FAST0899
 FAST0900
 FAST0901
 FAST0902
 FAST0903
 FAST0904
 FAST0905
 FAST0906
 FAST0907
 FAST0908
 FAST0909
 FAST0910
 FAST0911
 FAST0912
 FAST0913
 FAST0914
 FAST0915
 FAST0916
 FAST0917
 FAST0918
 FAST0919
 FAST0920
 FAST0921
 FAST0922
 FAST0923
 FAST0924
 FAST0925
 FAST0926
 FAST0927
 FAST0928
 FAST0929
 FAST0930
 FAST0931
 FAST0932
 FAST0933
 FAST0934
 FAST0935
 FAST0936
 FAST0937
 FAST0938
 FAST0939

```

      GO TO 165
      150 DO 160 J=1,MAX
      160 SG5(J,MM) = SG5(J,MM) + GST*X5E(J,K,L)
      165 CONTINUE
      IF(KMAX,LE,0)GO TO 170
      C INELASTIC NEXT
      DO 185 K=1,KMAX
      MM1= 101(K,M)
      MAX = JMX(K+3)
      IF(MM1,GT,0) GO TO 170
      MM = NN
      NN = NN + MINO(NTRANS,NEMOD-K)
      MM1 = MM
      GO TO 173
      170 MM = MM1/1000
      173 101(K,M) = 1000*MM + MAX0(MAX,MM1-1000*MM)
      IF((MM + MAX).LE,MHNC0) GO TO 175
      NERROR = NERROR + 1
      GO TO 185
      175 DO 180 J=1,MAX
      180 SG5(J,MM) = SG5(J,MM) + GST*X5I(J,K)
      185 CONTINUE
      190 CONTINUE
      200 CONTINUE
      390 CONTINUE
      NNMAX = NN - 1
      IF(LABEL(1),GE,0) WRITE(M2,200C)NNMAX,INI
      2000 FORMAT(1X,91M),3ANSCATTERING CROSS SECTIONS REQUIRED,16,3H OF,
      116,20H AVAILABLE LOCATIONS,19(1H*)
      IF(ING,LE,0) GO TO 210
      DO 510 I=1,NNMAX
      DO 500 J=1,NGMIN,8
      MAX = MINO(J + 7,NNMIN)
      IF(LABEL(1),GE,0) WRITE(M2,3000)I,(SGT(K,I),K=J,MAX)
      3000 FORMAT(1X,6H5IGNAT,15,1PBE12.4)
      500 CONTINUE
      510 CONTINUE
      MAXR = NIMAX + 1*MMAX
      DO 530 I=1,MAXR
      K = I
      IF(K,GT,NIMAX) K = K - NIMAX
      DO 520 J=1,NEMOD,8
      MAX = MINO(J + 7,NNMOD)
      IF(LABEL(1),GE,0) WRITE(M2,3010)K,(EAC(L,I),L=J,MAX)
      3010 FORMAT(1X,6HE-DUMP,15,1PBE12.4)
      520 CONTINUE
      530 CONTINUE

```

FAST0940
 FAST0941
 FAST0942
 FAST0943
 FAST0944
 FAST0945
 FAST0946
 FAST0947
 FAST0948
 FAST0949
 FAST0950
 FAST0951
 FAST0952
 FAST0953
 FAST0954
 FAST0955
 FAST0956
 FAST0957
 FAST0958
 FAST0959
 FAST0960
 FAST0961
 FAST0962
 FAST0963
 FAST0964
 FAST0965
 FAST0966
 FAST0967
 FAST0968
 FAST0969
 FAST0970
 FAST0971
 FAST0972
 FAST0973
 FAST0974
 FAST0975
 FAST0976
 FAST0977
 FAST0978
 FAST0979
 FAST0980
 FAST0981
 FAST0982
 FAST0983
 FAST0984
 FAST0985
 FAST0986

```

      210 IF(IN4,GT,0)CALL READIS(IN4,MTL)
      IF(INS,LE,0)GO TO 250
      IF(NUNIT0,EG,0)GO TO 230
      FST = 0.6025/1.00797
      DO 220 I=1,NRMAX
      220 RHO(I) = RHO(I)/FST
      230 CALL READIS(INS,RHO)
      IF(NUNIT0,EG,0)GO TO 250
      DO 240 I=1,NRMAX
      240 RHO(I) = FST*RHO(I)
      250 RETURN
      END
      SIDBTC ASKFOR M94/2,XRT
      RESULT*RESPONSE FUNCTION AND DETECTOR INPUT FOR FASTER*J.M.JORDAN*WAM
      SUBROUTINE RESULT(L1,L2,NTG,ELF,FGW,TDS,RSP,IDR,IDS,VOL,CDT,XDT,
      1,ELL,MTL,MOD,FAC,L3)
      COMMON/LIMITS/NSTORE,NERROR,NSPAX,NMAX,NRMAX,NBMAX,NSTMAX,
      1 NEMAX,NVMAX,NXMAX,NXMAX,NXMAX,NEMOD,NXSECT,NUNITO,
      2 NUNITX,NIMAX,NMAX,NORDER,NODWN,INELAS,NTRANS,
      3 NMAX,NCRAX,NRMAX,NVMOD,NCMAX,NLMAX,NTHAX,
      4 NDMAX,NMOD,NMOMENT,NMOD,NPDMAX,NPDMOD,NSDMAX,
      5 NVDMAX,NVDMOD,NPOINT,MODEL,MGDELQ,MODELQ,MODELV,
      6 NPRINT,NUNITS,NUMBER,KALIDE
      COMMON M(1)
      COMMON/REGSCA/NSRMAX,INSR
      COMMON/INDEXS/NDUNI25,IIDV,MDUM(47)
      DIMENSION MTL(1),RHO(1),EAC(L3,L2)
      DIMENSION NTG(1),ELF(1),FGW(1),TDS(3,1),RSP(1,L2),IDR(1),IDS(1),
      1 VOL(1),CDT(3,1),XDT(3,1),ELL(1)
      DIMENSION ADM(7),BDM(3),IDM(3)
      COMMON/TAPEID/M1
      COMMON/INPUTS/INI,IN2,IN3,IN4,IN5,IN6,IN7,IN8,IN9,IN10,IN11(14)
      IF(NGMAX,LT,NEMAX)GO TO 20
      DO 10 I=1,NEMAX
      10 NTG(I) = 1
      20 IF(IN2,GT,0) CALL READI(NEMAX,NTG)
      30 ELF(I) = ELL(I)
      DO 40 I=1,NEMAX
      40 ELL(I+1) = ELL(I+1)
      DO 50 I=1,NGMAX
      50 FGW(I) = ELF(I) - ELF(I+1)
      60 IF(INS,LE,0)GO TO 120
      DO 110 N=1,IN3
      CALL READF(IDN,BDM,ADM)
      I = IDM(I)
      DO 70 J=1,3

```

FAST0987
 FAST0988
 FAST0989
 FAST0990
 FAST0991
 FAST0992
 FAST0993
 FAST0994
 FAST0995
 FAST0996
 FAST0997
 FAST0998
 FAST0999
 FAST1000
 FAST1001
 FAST1002
 FAST1003
 FAST1004
 FAST1005
 FAST1006
 FAST1007
 FAST1008
 FAST1009
 FAST1010
 FAST1011
 FAST1012
 FAST1013
 FAST1014
 FAST1015
 FAST1016
 FAST1017
 FAST1018
 FAST1019
 FAST1020
 FAST1021
 FAST1022
 FAST1023
 FAST1024
 FAST1025
 FAST1026
 FAST1027
 FAST1028
 FAST1029
 FAST1030
 FAST1031
 FAST1032

```

70 TOS(J,1) = BDM(J)
IF(IDM(2).GE.0) GC TO 75
J = -IDM(2)
N = MTL(J) + NIMAX
RSP(1,1) = RHO(J)*EAC(1,1)
IF(M.GT.NIMAX) RSP(1,1) = RSP(1,1) + EAC(1,1)
DO 11 K=1,NEMAX
L = NTG(K)
RSP(L+1,1) = RHO(J)*EAC(K+1,1)
IF(M.GT.NIMAX) RSP(L+1,1) = RSP(L+1,1) + EAC(K+1,1)
11 CONTINUE
IF(LABEL(1+NGMOD/7).GE.0) WRITE(M2,2000)(RSP(J,1),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,1) = ADM(J+1)
IF(NGMOD.GT.5)CALL READE(NGMOD-5,RSP(6,1))
IF(IDM(2).FC.0)GO TO 100
DO 90 J=1,NGMOD
90 RSP(J,1) = ELF(J)*RSP(J,1)
95 CONTINUE
100 DO 110 J=1,NGMOD
110 RSP(J,1) = ADM(1)*RSP(J,1)
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(1) = IDM(2)
IDS(1) = IDM(3)
VOL(1) = ADM(1)
DO 140 J=1,3
CDT(J,1) = ADM(J+1)
140 XDT(J,1) = ADM(J+4)
150 FST = VECTOR(BDM,CDT(1,1),CDT(1,1))
552 NPDMAX = 0
NVDMAX = 0
DO 556 I=1,NVDMAX
IGIDR(I).GT.0)GO TO 554
NPDMAX = NPDMAX + 1
GO TO 556
554 IF(IDS(1).LE.0)NVDMAX = NVDMAX + 1
556 CONTINUE
NVDMOD = NVDMAX - NPDMAX

NSDMAX = NVDMOD - NVDMAX
NPDMOD = NPDMAX
IF(IN5.GT.0) CALL READI(NVDMOD,I(IIDV))
IF(IN6.GT.0) CALL READI(NSRMAX,H(1))
RETURN
END
$BFTC RANDOM M94/2,XRT
CRANDOM+SAMPLING PARAMETER INPUTS FOR PROGRAM FASTER*Y.M.JORDAN*MANL*66*
SUBROUTINE RANDOM(RSI,ALP,VMD,GIM,ATM,ALM,ALH,NPC,VEE,L1,L2)
DIMENSION NPC(5,1),VEE(4,1,2,1),RAT(5)
DIMENSION RSI(1),ALP(5,1),VMD(5,1),GIM(1),ALM(1),ALH(1)
COMMON/TAPEID/H1 ,H2
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/NSDIRE,NERROR,NSMAX ,NAMAX ,NRMAX ,NBMAX ,NSTMAX ,
3 NEMAX ,NPMAX ,NMMAX ,NEMAX,NEMOD ,NNSCT,NUNITO ,
4 NUNITX,NIMAX ,NMPAX ,NORDER,NDOWN ,INELAS,NTRANS ,
5 NNMAX ,NGMAX ,NEMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
6 NMAX ,NGMOD ,NMOMENT ,NDMOD ,NPDMAX ,NPDMOD ,NSDMAX ,
7 NVDMAX ,NVDMOD ,NPCINT ,MODELP ,MODELO ,MODELU ,MODELV ,
8 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(1,1))
CALL READE(5,RAT)
DO 40 J=1,5
ALP(J,1) = 0.0
K = NPC(J,1)
IF(K.EQ.1)GO TO 40
ALP(J,1) = -ALOG(RAT(J))
1 J = JMAX1(VMD(J,1)-VEE(1,J,1),VEE(K,J,1)-VMD(J,1))
40 CONTINUE
560 CONTINUE
793 IF(IN5.GT.0)CALL READE(NEMAX,GIM)
IF(IN6.GT.0)CALL READE(NEMAX,ALM)
IF(IN7.LE.0)GO TO 60

```

FAST1034
 FAST1035
 FAST1036
 FAST1037
 FAST1038
 FAST1039
 FAST1040
 FAST1041
 FAST1042
 FAST1043
 FAST1044
 FAST1045
 FAST1046
 FAST1047
 FAST1048
 FAST1049
 FAST1050
 FAST1051
 FAST1052
 FAST1053
 FAST1054
 FAST1055
 FAST1056
 FAST1057
 FAST1058
 FAST1059
 FAST1060
 FAST1061
 FAST1062
 FAST1063
 FAST1064
 FAST1065
 FAST1066
 FAST1067
 FAST1068
 FAST1069
 FAST1070
 FAST1071
 FAST1072
 FAST1073
 FAST1074
 FAST1075
 FAST1076
 FAST1077
 FAST1078
 FAST1079
 FAST1080

 FAST1081
 FAST1082
 FAST1083
 FAST1084
 FAST1085
 FAST1086
 FAST1087
 FAST1088
 FAST1089
 FAST1090
 FAST1091
 FAST1092
 FAST1093
 FAST1094
 FAST1095
 FAST1096
 FAST1097
 FAST1098
 FAST1099
 FAST1100
 FAST1101
 FAST1102
 FAST1103
 FAST1104
 FAST1105
 FAST1106

 FAST1107
 FAST1108
 FAST1109
 FAST1110
 FAST1111
 FAST1112
 FAST1113
 FAST1114
 FAST1115
 FAST1116
 FAST1117
 FAST1118
 FAST1119
 FAST1120
 FAST1121
 FAST1122
 FAST1123
 FAST1124
 FAST1125
 FAST1126

```

      CALL READE(NEMAX,ALM)
      DO 50 I=1,NEMAX
50  ALM(I) = 0.5*ALOG(ALM(I))
80  IF(INP.LE.0)GO TO 80
      CALL READE(NEMAX,ALH)
      DO 70 I=1,NEMAX
70  ALH(I) = 0.5*ALOG(ALH(I))
80  IF(INP.LE.0)GO TO 9C
      CALL READE(18,ATA)
      ATA = ALOG(ATA)
      ATB = -ALOG(ATB)/3.141596
90  RETURN
      END
$1BFTC IREAD M94/2,XR7
CREADI *INTEGER INPUT, 2413 FORMAT
SUBROUTINE READI(MAX,ION)
COMMON/TAPEID/M1,M2
DIMENSION IOM(10000),H(2)
DO 30 I=1,MAX,24
  MOD = I + 23
  IF(MOD.GT.MAX)GO TO 10
  READ(M1,1000)(IOM(J),J=1,MOD),H
  GO TO 20
10  JMX = MOD - MAX
  MOD = MAX
  READ(M1,1000)(IOM(J),J=1,MOD),(K,J=1,JMX),H
20  IF(LABEL(1).GE.0)WRITE(M2,2000)H,(IOM(J),J=1,MOD)
30  CONTINUE
1000 FORMAT(24I3,2A4)
2000 FORMAT(1X,2A4,3H...,24I4)
      RETURN
      END
$1BFTC IREAD M94/2,XR7
CREADI *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=1,MOD),H
  GO TO 20
10  JMX = MOD - MAX
  MOD = MAX
  READ(M1,1000)(ADM(J),J=1,MOD),(X,J=1,JMX),H
20  IF(LABEL(1).GE.0)WRITE(M2,2000)H,(ADM(J),J=1,MOD)
30  CONTINUE
1000 FORMAT(8E9.0,2A4)
2000 FORMAT(1X,2A4,3H...,1P8E12.4)
      RETURN
      END
$1BFTC IREAD 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=1,MOD),H
  GO TO 20
10  JMX = MOD - MAX
  MOD = MAX
  READ(M1,1000)(ADM(J),J=1,MOD),(X,J=1,JMX),H
20  IF(LABEL(1).GE.0)WRITE(M2,2000)H,(ADM(J),J=1,MOD)
30  CONTINUE
1000 FORMAT(18A4)
2000 FORMAT(1X,2A4,27I1H.),18A4)
      RETURN
      END
$1BFTC I$READ M94/2,XR7
SUBROUTINE READIS(MAX,ION)
DIMENSION IOM(10000),JOM(2,12)
DO 10 I=1,MAX,12
  MOD = MIN(12,MAX+1-I)
  CALL READI(2*MOD,JOM)
  DO 10 J=1,MOD
    K = JOM(1,J)
    IOM(K) = JOM(2,J)
  GO TO 20
10  IOM(K) = JOM(2,J)
      RETURN
      END
$1BFTC I$READ M94/2,XR7
SUBROUTINE READIS(MAX,ION,ADM)
COMMON/TAPEID/M1,M2
DIMENSION IOM(10000),ADM(10000),H(2)
DO 30 I=1,MAX,6
  MOD = I + 5
  IF(MOD.GT.MAX)GO TO 10
  READ (M1,1000)(IOM(J),ADM(J),J=1,MOD),H
  GO TO 20
10  JMX = MOD - MAX
  MOD = MAX
  READ(M1,1000)(IOM(J),ADM(J),J=1,MOD),(K,X,J=1,JMX),H
20  IF(LABEL(1).GE.0)WRITE(M2,2000)H,(IOM(J),ADM(J),J=1,MOD)

```

FAST1127
FAST1128
FAST1129
FAST1130
FAST1131
FAST1132
FAST1133
FAST1134
FAST1135
FAST1136
FAST1137
FAST1138
FAST1139
FAST1140
FAST1141
FAST1142
FAST1143
FAST1144
FAST1145
FAST1146
FAST1147
FAST1148
FAST1149
FAST1150
FAST1151
FAST1152
FAST1153
FAST1154
FAST1155
FAST1156
FAST1157
FAST1158
FAST1159
FAST1160
FAST1161
FAST1162
FAST1163
FAST1164
FAST1165
FAST1166
FAST1167
FAST1168
FAST1169
FAST1170
FAST1171
FAST1172
FAST1173

FAST1174
FAST1175
FAST1176
FAST1177
FAST1178
FAST1179
FAST1180
FAST1181
FAST1182
FAST1183
FAST1184
FAST1185
FAST1186
FAST1187
FAST1188
FAST1189
FAST1190
FAST1191
FAST1192
FAST1193
FAST1194
FAST1195
FAST1196
FAST1197
FAST1198
FAST1199
FAST1200
FAST1201
FAST1202
FAST1203
FAST1204
FAST1205
FAST1206
FAST1207
FAST1208
FAST1209
FAST1210
FAST1211
FAST1212
FAST1213
FAST1214
FAST1215
FAST1216
FAST1217
FAST1218
FAST1219
FAST1220

```

30 CONTINUE
1000 FORMAT(13I9,0I,2A4)
2000 FORMAT(1X,2A4,3H...6I14,1PE12.4)
RETURN
END
SUBFTC ESREAD N94/2,XR7
SUBROUTINE HEADS(MAX,ADM)
DIMENSION ADM(10000),IDM(6),BDM(6)
DO 10 I=1,MAX,6
MOD = MIN(6,MAX-I-1)
CALL READI(MOD,IDM,BDM)
DO 10 J=1,MOD
K = IDM(IJ)
10 ADM(K) = BDM(IJ)
RETURN
END
SUBFTC EEREAD N94/2,XR7
CREADEE=ALTERNATING FLOATING INPUT, 0E9.0 FORMAT
SUBROUTINE READEE(MAX,ADM,BDM)
DIMENSION ADM(10000),BDM(10000),CDM(2,4)
K = 0
DO 10 I=1,MAX,4
MOD = MIN(4,MAX-K)
CALL READEE(2*MOD,CDM)
DO 10 J=1,MOD
K = K + 1
ADM(K) = CDM(1,J)
10 BDM(K) = CDM(2,J)
RETURN
END
SUBFTC SREAD N94/2,XR7
CREADS = SURFACE FORMAT, 3I3,7E9.0
SUBROUTINE READS(IDM,ADM)
COMMON/TAPEID/M1,M2
DIMENSION IDM(3),ADM(7),H(2)
READ(M1,1000)IDM,ADM,H
IF(LABEL(1),GE.0)WRITE(M2,2000)H,IDM,ADM
1000 FORMAT(3I3,7E9.0,2A4)
2000 FORMAT(1X,2A4,3H...3I4,1PE12.4)
RETURN
END
SUBFTC RREAD N94/2,XR7
CREADR = VOLUME SOURCE FORMAT, 4E9.0,12I3 FORMAT
SUBROUTINE RREAD(IDM,ADM)
COMMON/TAPEID/M1,M2
DIMENSION ADM(4),IDM(12),H(2)
READ(M1,1000)IDM,ADM,H

```

FAST1221
FAST1222
FAST1223
FAST1224
FAST1225
FAST1226
FAST1227
FAST1228
FAST1229
FAST1230
FAST1231
FAST1232
FAST1233
FAST1234
FAST1235
FAST1236
FAST1237
FAST1239
FAST1240
FAST1241
FAST1242
FAST1243
FAST1244
FAST1245
FAST1246
FAST1247
FAST1248
FAST1249
FAST1250
FAST1251
FAST1252
FAST1253
FAST1254
FAST1255
FAST1256
FAST1257
FAST1258
FAST1259
FAST1260
FAST1261
FAST1262
FAST1263
FAST1264
FAST1265
FAST1266
FAST1267
FAST1268

```

IF(LABEL(1),GE.0)WRITE(M2,2000)H,IDM,ADM
1000 FORMAT(12I3,4E9.0,2A4)
2000 FORMAT(1X,2A4,3H...12I4,1PE12.4)
RETURN
END
SUBFTC FREAD N94/2,XR7
CREADF = RESPONSE FUNCTION FORMAT, 2I3,3A4,6E9.0
SUBROUTINE READF(IDM,ADM,BDM)
COMMON/TAPEID/M1,M2
DIMENSION IDM(2),ADM(3),BDM(6),H(2)
READ(M1,1000)IDM,ADM,BDM,H
IF(LABEL(1),GE.0)WRITE(M2,2000)H,IDM,ADM,BDM
1000 FORMAT(2I3,3A4,6E9.0,2A4)
2000 FORMAT(1X,2A4,3H...2I4,2X,3A4,2X,1PE12.4)
RETURN
END
SORIGIN ALPHA,REW
SUBFTC RSOLVE N94/2,XR7
SUBROUTINE SOLVIT
COMMON/LIMITS/NSSTORE,NERROR,NSMAX,NMAX,NRMAY,NBMAX,NHSTMAX,
1 NEMAX,NVMAX,NKMAX,NZEMAX,NEMOD,NKSECT,NUNITD,
2 NUNITX,NIMAX,NMPAX,NORDER,NODOWN,NTELAS,NTRANS,
3 NNMAY,NHGMAY,NFMAY,NVMOD,NCMAX,NLMAY,NIMAX,NIMAX,
4 NDMAX,NHMOD,NKMOD,NMOD,NPDHAX,NPHMOD,NSDMAX,
5 NYDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,
6 NPRINT,NUNITS,NUMBER,KALOE
COMMON/INDEXS/INTP,IAZ,ISV,INTL,IRHO,IXR,IELL,
1 IELM,IAS,IDE,INSG,INPC,IJSN,IJSX,
2 ITR,ISUV,IATN,IATM,IESB,ISSH,IDEN,
3 INTG,IELF,IFGW,ITDS,IIOV,IIDR,IIDS,
4 IVOL,ICDT,IXOT,IRSI,ALP,IIVD,IOLP,
5 IATN,IALM,IALH,IA,INS,IIVE,IVAL,
6 ISPW,ISPE,IATD,IRSP,ISGT,IIDE,IIDI,
7 ISGS,NEXT,IXMP,ISGR,IWS,IES,IWC,
8 IEC,IWD,IU,IW,IW,IST,IST,
9 INRG,INSC,ISTP,INRP,INCP,IFXP,IFXS,
1 IFXT,IFXE,IFXA
COMMON H(1)
IF(NPOINT,GT.0)GO TO 20
CALL SUBR
1(NGMAY,1,H(INS),H(IES),H(IMC),H(IEC),H(IST),H(INRG),H(IML),
2 H(IDR),H(IIS),H(IIDV),H(IXDT),H(ICDT),H(IPXP),H(IPXS),H(INRP),
3 H(INSC),H(IALM),H(IALH),H(IRHO),H(IELL),H(ISSH),H(IGT),H(ENOD)
GO TO 10
20 CALL SOLVER(1,NEMOD,NGMAY,H(IXDT),H(ICDT),H(IST),H(INRG),H(INSC),
1 H(IML),H(IRHO),H(ISSH),H(IGT),H(IELL),H(IWS),H(IES),H(IST),H(
2 H(IMC),H(IEC),H(IPXP),H(IALM),H(IALH),H(IPXP),H(IPXS),H(IST),H(

```

FAST1269
FAST1270
FAST1271
FAST1272
FAST1273
FAST1274
FAST1275
FAST1276
FAST1277
FAST1278
FAST1279
FAST1280
FAST1281
FAST1282
FAST1283
FAST1284
FAST1285
FAST1286
FAST1287
FAST1288
FAST1289
FAST1290
FAST1291
FAST1292
FAST1293
FAST1294
FAST1295
FAST1296
FAST1297
FAST1298
FAST1299
FAST1300
FAST1301
FAST1302
FAST1303
FAST1304
FAST1305
FAST1306
FAST1307
FAST1308
FAST1309
FAST1310
FAST1311
FAST1312
FAST1313
FAST1314

[illegible]

```

1 IF(TOTALN.EC.O)GO TO 13
2 EBAR = TOTALT/TOTALN
3 DO I2 I=1,NEMAX
4   IF(EBAR.GE.EL(1+I))GO TO 14
5 CONTINUE
6 I3 = (JMIN + JMAX)/2
7 JBAR = I
8 POT = USTAR(ISTC,ODC,X+C,NST,ST,NRG)
9 IF(NST.EQ.ODGT TO 220
10  RUNTOT = RUNTOT + STC
11  NN = NRG(IST)
12  POT = POT/STC**2
13  CALL KERNEL(1,NST,ST,NRG)
14  DO 20 I=1+3
15    X(I) = X(I) + C(I)*STC
16    C(I) = C(I)
17  KMIN = JMIN
18  KMAX = JMAX
19  DO 30 I=JMIN,JMAX
20    WCI() = POT**S(I)*EXP(-XMP(I))
30 EC(II) = ES(I)
40 NTAUPL = NTAULL
50 NTALLY = 0
60 TOTALN = 0.0
70 TOTALF = 0.0
80 IF(NPDMAX.EC.O) GC TO 50
90 NPD = 0
100 DO I=1,NEMAX
110 IF(IDR(II).GT.O)GO TO 80
120 NPD = NPD + 1
130 STM = VECTORIX,XD(II, 1),C1
140 IF(KKK.GT.1)GO TO 50
150 CALL SZERC(NN,PCT,X,C)
160 GO TO 100
170 CALL SINGLE(NN,COSINE(C,CC1),.0)
180 IF(IJMIN.GT.JMAX)GC TO 70
190 TOT = RUNTOT + STP
200 CALL PATH(NN,STM,X,C,NST,ST,NRG,NSC)
210 CALL KERNEL(1,NST,ST,NRG)
220 CALL DETECT(II=-1,.0/ST**2,COSINE(C,CDT(1,1)),0.0)
230 IF(NPND.EQ.NPDMAX)GO TO 90
240 CONTINUE
250 IF(KKK.LT.KALDIE) GC TO 95
260 IF(NVDMOD.EC.O) GC TC 210
270 IF(KKK.GT.O) GC TO 102
280 PDX = POT
290 IF(MODELO.GT.O) GC TO 100

```

```

      MGT = 0
      GO TO 108
100 FST = VECTOR(XCT,X,CC)
      JBAR = JZERO
      GO TO 104
102 PDX = 1.0
      MGT = 1
104 FST = RMD(INN)*SSH(JBAR)
      GST = 0.0
      M = MTL(INN)
      IF(M.GT.0) GST = SGT(JBAR,M)
      ALPHA = 0.0
      NST = FST + GST
      IF(NST.GT.0.0) ALPHA = (FST*ALM(JBAR) + GST*ALM(JBAR))/NST
      STM = VECTOR(X,BCC,CP)
      BETA = 0.0
      IF(FST.LE.DELTA) GO TO 108
      CALL PATH(INN,FST-DELTA,X,CP,NST,ST,NRG,NSC)
      XMPT = 0.0
      DO 106 I=1,NST
        J = NRG(I)
        K = MTL(J)
        SIG = RMD(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=1,10MEGA
      IF(MGT.GT.0) GO TO 110
      PDA = PDX*OSTAR(INN,X,C)
      GO TO 112
110 PDA = PDX*VSTAR(CP,CC,C,BETA,BS*ALPHA)
112 IF(LLL.LT.10MEGA) PCA = PDA/FLOAT(NUMBER)
      IF(KKK.GT.1) GO TO 114
      CALL SZERC(INN,PIA,X,C)
      GO TO 116
114 CALL SINGLE(INN,COSINE(C,CC),PDA)
116 IF(JMIN.GT.JMAX) GO TO 202
      CALL PATH(INN,1.0E+30,X,C,NST,ST,NRG,NSC)
      IF(LLL.EQ.10MEGA) GO TO 205
      NSTMIN = 1
      STT = 0
      TOT = RUNTOT
      DO 200 N=1,NST
        L = NRG(IN)
        M = MTL(L)
        IF(NVDMAX.EQ.0) GO TO 150

```

```

FAST1409
FAST1410
FAST1411
FAST1412
FAST1413
FAST1414
FAST1415
FAST1416
FAST1417
FAST1418
FAST1419
FAST1420
FAST1421
FAST1422
FAST1423
FAST1424
FAST1425
FAST1426
FAST1427
FAST1428
FAST1429
FAST1430
FAST1431
FAST1432
FAST1433
FAST1434
FAST1435
FAST1436
FAST1437
FAST1438
FAST1439
FAST1440
FAST1441
FAST1442
FAST1443
FAST1444
FAST1445
FAST1446
FAST1447
FAST1448
FAST1449
FAST1450
FAST1451
FAST1452
FAST1453
FAST1454
FAST1455

```



```

      DO 140 I=1,NOMAX
      IF(10R(I).NE.L) GO TO 140
      IF(10S(I).NE.0) GO TO 140
      ANG = COSINE(C,COT(1,1))
      IF(M.LT.0) GO TO 120
      FST = 1.0/VCL(I)
      RHUM = RMD(L)
      GO TO 130
120 FST = ST(I)/VCL(I)
      TOT = TOT + ST(I)/2.0
130 CALL KERNEL(NSTMIN,N-1,ST,NRG)
      CALL DETECT(I,M,FST,ANG,ST(I))
      IF(M.LT.0) TOT = TOT - ST(I)/2.0
      NSTMIN = N + 1
      GO TO 150
140 CONTINUE
150 STT = STT + ST(I)
      TOT = RUNTOT + STT
      K = NSC(IN)
      IF(K*NSOMAX.LE.0) GO TO 200
      MAX = 2 - N/NST
      NGT = 0
      DO 190 N=1,MAX
      DO 160 I=1,NOMAX
      IF(10R(I).EQ.L.AND.10S(I).EQ.K) GO TO 170
160 CONTINUE
      GO TO 190
170 IF(INGT.GT.0) GO TO 180
      NGT = 1
      CALL NORMAL(K,STT,X,C,CN)
      ANG = COSINE(C,CA)
      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(IN)
200 CONTINUE
202 IF(LLL.EQ.10MEGA) GO TO 220
205 CONTINUE
210 CONTINUE
220 CONTINUE
      DO 230 I=1,NOMAX
      DO 230 J=1,NOMAX
      FXS(J,I) = FXS(J,I) + FXP(J,I)**2
230 FXP(J,I) = 0.0
235 CONTINUE
240 CONTINUE

```

```

FAST1456
FAST1457
FAST1458
FAST1459
FAST1460
FAST1461
FAST1462
FAST1463
FAST1464
FAST1465
FAST1466
FAST1467
FAST1468
FAST1469
FAST1470
FAST1471
FAST1472
FAST1473
FAST1474
FAST1475
FAST1476
FAST1477
FAST1478
FAST1479
FAST1480
FAST1481
FAST1482
FAST1483
FAST1484
FAST1485
FAST1486
FAST1487
FAST1488
FAST1489
FAST1490
FAST1491
FAST1492
FAST1493
FAST1494
FAST1495
FAST1496
FAST1497
FAST1498
FAST1499
FAST1500
FAST1501
FAST1502

```



```

CALL ANSWER(NNN)
250 CONTINUE
RETURN
END
$IBTC SOLVE #94/2,XR7
CSOLVER-POINT SOLUTION OF THE BOLTZMANN EQUATION* T.M.JORDAN*MANL*1966*
SUBROUTINE SOLVER(L1,L2,L3,XDT,CDT,ST,NRG,NSC,MTL,RHO,SSH,SGT,ELL,
1MS,ES,MC,EC,XMP,ALM,ALH,FAP,FXS,STP,NRP,NGP,IDR1)
DIMENSION XT(3),CT(3),CT(3),X(3),CG(3),CDI(3),XP(3)
DIMENSION XDT(3),L1,CDT(3),L1,ST(1),NRG(1),MTL(1),RHO(1),SSH(1),
1 SGT(L2,L1),ELL(1),WS(1),ES(1),MC(1),EC(1),XMP(1),ALM(1),
2 ALH(1),FAP(L3,L1),FXS(L3,L1),IDR1(1)
COMMON/TAPE10/M1,M2
COMMON/LIMITS/NSTORE,NERROR,NSMAX,NAMAX,NRMAX,NBMAX,NSTMAX,
1 NEMAX,NVMAX,NKPMAX,NKEMAX,NEMOD,NKSECT,NUNITO,
2 NUNITX,NHMAX,NHMAX,NHORDER,NHODDN,INELAS,NTRANS,
3 NMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,
4 NMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,
5 NMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,
6 NPRINT,NUNITS,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,NHMAX,
COMMON/FLUXES/KKK,NMALLY,ERROR,TCTALN,TOTALE,RHON,SNORM,TOTEST,
COMMON/POINTS/NTOTAL,I11,MOM
COMMON/OTHERS/RADIUS,XCT(3),DELTA,BDC(3),ATA,ATB,ATC,
1 ATD,AT,AT,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 I=1,NHMAX
NTOTAL = 0
IF(IDR1(I),GT,0) GO TC 27
NPDRUN = NPDRUN + 1
DO 1 I=1,3
XT(I) = XDT(I,I11)
1 CT(I) = CDT(I,I11)
CALL GROUP(XT)
SHORR = AMAX1(100,0,VECTOR(XCT,XT,CT))
DO 30 NMM=1,NPRINT
DO 29 NMM=1,NUNITS
IF(MODELP,GT,0) GC TC 53
NHALF = 1
POS = PSTAR(N,X1)
IF(NN,EQ,0) GO TO 26
GO TO 94
93 CONTINUE

```

FAST1503
FAST1504
FAST1505
FAST1506
FAST1507
FAST1508
FAST1509
FAST1510
FAST1511
FAST1512
FAST1513
FAST1514
FAST1515
FAST1516
FAST1517
FAST1518
FAST1519
FAST1520
FAST1521
FAST1522
FAST1523
FAST1524
FAST1525
FAST1526
FAST1527
FAST1528
FAST1529
FAST1530
FAST1531
FAST1532
FAST1533
FAST1534
FAST1535
FAST1536
FAST1537
FAST1538
FAST1539
FAST1540
FAST1541
FAST1542
FAST1543
FAST1544
FAST1545
FAST1546
FAST1547
FAST1548
FAST1549

```

POS = 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
RUNITO = 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,POS,X,C1)
GO TO 4
3 CALL SINGLE(NN,COSINE(C,CC),L,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 = RUNITO + STD
CALL DETECT(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(1+I)) GO TO 8
6 CONTINUE
7 T = JMIN
8 JBAR = 1
NGT = 0
XMP = 0.0
STW = STD/2.0
STT = 0.0
SIGN = 0.0
DO 9 I=1,NST
J = NRG(I)
SIG = RHO(J)*SSH(JBAR)
SIGN = SIGN + SIG*ST(I)
K = MTL(J)
IF(K,GT,0)SIG = SIG + SGT(JBAR,K)

```

FAST1550
FAST1551
FAST1552
FAST1553
FAST1554
FAST1555
FAST1556
FAST1557
FAST1558
FAST1559
FAST1560
FAST1561
FAST1562
FAST1563
FAST1564
FAST1565
FAST1566
FAST1567
FAST1568
FAST1569
FAST1570
FAST1571
FAST1572
FAST1573
FAST1574
FAST1575
FAST1576
FAST1577
FAST1578
FAST1579
FAST1580
FAST1581
FAST1582
FAST1583
FAST1584
FAST1585
FAST1586
FAST1587
FAST1588
FAST1589
FAST1590
FAST1591
FAST1592
FAST1593
FAST1594
FAST1595
FAST1596

```

XMP1 = XMP1 + SIG*ST(I)
IF(NGT.EQ.1)GO TO 9
STY = STI + ST(I)
IF(STI.LT.STH)GO TO 9
XMPH = XMP1 - SIG*(STI - STH)
NGT = 1
9 CONTINUE
BETA = 0.0
IF(NHALF.EQ.2)XMPH = XMP1 - XMPH
IF(XMP1.GT.0.0) BETA=AS*(ALOG(12.0*XMP1)/4.0
IF(MODELV.EQ.2) BETA = AS*XMP1/2.0
NNC = NN
IF(KKK*MODELQ1.EC.1) GO TO 10.
IF(XMP1.EQ.0.0)GO TO 1C
IF(XMPH.LT.XMP1)GO TO 11
10 PDF = 1.0
NGT = 0
GO TO 12
11 IF(XMPH.EQ.0.0)GO TO 14
PHALF = XMPH/XMP1
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)
BDC(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)
BDC(I) = X(I)
17 X(I) = XT(I)
NN = NZERO
18 ALPHA = 0.0
IF(XMP1.GT.0.0)ALPHA=(STGH*ALH(JBAR)+(XMP1-SIGH)*ALH(JBAR))/XMP1
IF(KKK*MODELQ1.GT.1) GO TO 300
PDA = QSTAR(INN,X,C)
GO TO 301
300 PDA = VSTAR(ICD,CC,C,BETA,B5*ALPHA)

```

FAST1597
FAST1598
FAST1599
FAST1600
FAST1601
FAST1602
FAST1603
FAST1604
FAST1605
FAST1606
FAST1607
FAST1608
FAST1608
FAST1609
FAST1610
FAST1611
FAST1612
FAST1613
FAST1614
FAST1615
FAST1616
FAST1617
FAST1618
FAST1619
FAST1620
FAST1621
FAST1622
FAST1623
FAST1624
FAST1625
FAST1626
FAST1627
FAST1628
FAST1629
FAST1630
FAST1631
FAST1632
FAST1633
FAST1634
FAST1635
FAST1636
FAST1637
FAST1638
FAST1639
FAST1639
FAST1639

```

301 CONTINUE
STM = 1.0E+30
IF(NGT.EQ.0)GO TO 19
PSI = COSINE(C,CD)
IF(PSI.GT.0.0)STM = STM/PSI
19 CALL PATH(INN,STM,X,C,NST,ST,NRG,NSC)
PDT = USTAR(STC,BDC,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 STERDINN(PDS,X,C)
GO TO 22
21 CALL SINGLE(INN,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(INN,STM,X,C,NSTP,SEP,ARP,NCP)
CALL KERNEL(1,NSTP,STP,NRP)
201 CONTINUE
DO 23 I=JMIN,JMAX
EC(I) = ES(I)
23 WC(I) = PDX*WC(I)*EXP(-XMP(I))
KHIN = 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 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

```

FAST1639
FAST1640
FAST1641
FAST1642
FAST1643
FAST1644
FAST1645
FAST1646
FAST1647
FAST1648
FAST1649
FAST1650
FAST1651
FAST1652
FAST1653
FAST1654
FAST1655
FAST1656
FAST1657
FAST1658
FAST1659
FAST1660
FAST1661
FAST1662
FAST1663
FAST1664
FAST1665
FAST1666
FAST1667
FAST1668
FAST1669
FAST1670
FAST1671
FAST1672
FAST1673
FAST1674
FAST1675
FAST1676
FAST1677
FAST1678
FAST1679
FAST1680
FAST1681
FAST1682
FAST1683
FAST1684
FAST1685

```

      ON I01 I=1,3
101 BDC(I) = BDCUM(I)
      RETURN
      END
$1BFTC GROUP1 M94/2,XR7
SUBROUTINE GROUP1X1
COMMON H111
COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NMAX ,NRMAX ,NBMAX ,NSTMAX,
1 NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
2 NUNITX,NIMAX ,NMPAX ,NCRDER,NODOWN ,INELAS,NTRANS,
3 NNMIX ,NGMAX ,NFMIX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
4 NDMAX ,NGMOD ,NDPENT ,NDMOD ,NPDMAX ,NPDOD ,NSDMAX,
5 NVDMAX ,NVDMOD ,NPICINT ,MODELP ,MODELG ,MODELU ,MODELV,
6 APRINT,NUNITS,NUMBER,KALIDE
COMMON/INDEXS/INTP ,IAZ ,IISV ,IMTL ,IRHO ,IXR ,IELL ,
1 IELM ,IAE ,IBE ,IMSG ,INPC ,IJSN ,IJSX ,
2 IXTR ,ISUV ,IATN ,IATW ,IESD ,ISSH ,IDEN ,
3 INYG ,IELF ,IFGW ,ITDS ,IIDV ,IIDR ,IIDS ,
4 IVOL ,ICDT ,ICXT ,IRSI ,IALP ,IVMO ,IGIP ,
5 IAIM ,IALM ,IALH ,IA ,INS ,IVEE ,IVAL ,
6 ISPM ,ISPE ,IATD ,IRSP ,ISGT ,IIDE ,IIDL ,
7 ISGS ,NEXT ,IXMP ,ISGR ,IWS ,IES ,IWC ,
8 IEC ,IND ,IU ,IV ,IW ,ISI ,IST ,
9 INRG ,INSC ,ISTP ,INRP ,INCP ,IFXP ,IFXS ,
10 IFXT ,IFXE ,IFXA
. CALL GROOM(X1,NEMAX,H(IST),H(INRG),H(INS),H(IISV),
1 H(INS),H(IES),H(IJSN),H(IJSX),H(ISPW),H(ISPE),H(IXMP),H(IAIM),
2 H(IGIM),H(IELL))
      RETURN
      END
$1BFTC GROUP1 M94/2,XR7
SUBROUTINE GROOM(X,LL,L2,ST,NRG,NSC,ISV ,WS,ES,JSN,JSX,SPW,
1 ISPE,XMP,AIM,GIM,ELL)
COMMON/INDEXG/JZERO
COMMON/OTHERS/RADUS,XCT(3),DELTA ,BDC(3),ATA ,ATB ,ATC ,
1 ATO ,AT ,AT ,AS ,BS ,JMIN ,JMAX ,
2 KMIN ,KMAX ,JBAR ,NZERO
COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NMAX ,NRMAX ,NBMAX ,NSTMAX,
1 NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
2 NUNITX,NIMAX ,NMPAX ,NCRDER,NODOWN ,INELAS,NTRANS,
3 NNMIX ,NGMAX ,NFMIX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
4 NDMAX ,NGMOD ,NDPENT ,NDMOD ,NPDMAX ,NPDOD ,NSDMAX,
5 NVDMAX ,NVDMOD ,NPICINT ,MODELP ,MODELG ,MODELU ,MODELV,
6 APRINT,NUNITS,NUMBER,KALIDE
DIMENSION NRG(1),ISV(1) ,WS(1),ES(1),JSN(1),JSX(1),
1 ISPW(L2,L1),SPE(L2,L1),XMP(1),AIM(1),GIM(1),ELL(1) ,
2 DELTA(1)
      NZERO = LOCATE(NGS,X)
      STM = VECTOR(X,XCT,C)
      CALL PATHINZERO,STM,X,C,NST,ST,NRG,NSC)
      I = NST + 1
      DO 1 K=1,NST
      I = I - 1
      L = NRG(I)
      J = ISV(L)
      IF(J,GT,0) GO TO 2
1 CONTINUE
      JZERO = NEMOD/2
      GO TO 100
2 NTT = I
      C CALCULATE AVERAGE SOURCE GROUP INDEX
      TOTN = 0.0
      TOTE = 0.0
      MIN = JSN(J)
      MAX = JSX(J)
      JMIN = MIN
      JMAX = MAX
      DO 10 I=MIN,MAX
      WS(I) = SPW(I,J)
10 ES(I) = SPE(I,J)
      IF(NTT.EQ.1)GO TO 30
      CALL KERNEL(1,NTT-1,ST,NRG)
      IF(XMP(1),EQ.0.0)GO TO 30
      DO 20 I=MIN,MAX
      WS(I) = WS(I)*XMP(I)/(1.0 + AIM(I)*XMP(I))
20 WS(I) = WS(I)*XMP(I)/(1.0 + AIM(I)*XMP(I))
30 CALL KERNEL(1,0,ST,NRG)
      CALL KERNEL(NTT,NTT,ST,NRG)
      DO 40 I=MIN,MAX
      FST = GIM(I)*WS(I)
      IF(XMP(1),NE.0.0)FST = FST/XMP(1)
      TOTN = TOTN + FST
40 TOTE = TOTE + FST*ES(I)
      EBAR = TOTE/TOTN
      DO 50 I=MIN,MAX
      IF(EBAR,GE,ELL(1+1))GO TO 60
50 CONTINUE
      I = (MIN + MAX)/2
60 JZERO = I
100 RETURN
      END
$1BFTC SZERO M94/2,XR7
CSZERO *FIXED SOURCE EVALUATION FOR PROGRAM FASTER*Y.H. JORDAN*MAHL*1966*
SUBROUTINE SZERO(NNN,PDF,XXX,CCC)
DIMENSION XXX(1),CCC(1)

```

```

COMMON      H(1)
COMMON/DUMSZ/NN ,PCS ,X(3) ,C(3)
COMMON/LIMITS/NSICRE,NERROR,NSMAX ,NMAX ,NRMAX ,NMAX ,NSTMAX,
1  NEMAX ,NVMAX ,NXPAX ,NXMAX,NEMOD ,NXSECT,NUNITO,
2  NUNITX,NMAX ,NMAX ,NORDER,NODOWN ,INELAS,NTRANS,
3  NMAX ,NGMAX ,NFPAX ,NVMOD ,NCHMAX ,NLMAX ,NMAX ,
4  NMAX ,NGMOD ,NOPEN ,NMOD ,NPOMAX ,NPOMOD ,NSMAX ,
5  NVMAX ,NVOMOD ,NPCINT ,MODEL ,MODELQ ,MODELV,
6  NPRINT,NUNITS,NUMBER,KALIDE
COMMON/INDEXS/INTP ,IAZ ,IISV ,IMTL ,IRHO ,IXR ,IELL ,
1  IELA ,IAE ,IBE ,INSG ,INPC ,IJSN ,IJSX ,
2  IXTX ,ISUV ,IATN ,IATM ,IESB ,ISSH ,IDEM ,
3  INTG ,IELF ,IFGW ,ITDS ,IDV ,IDR ,IDOS ,
4  IYOL ,ICCT ,IXGT ,IRSI ,IALP ,IVMD ,IGIM ,
5  IAIM ,IALM ,IALH ,IA ,INS ,IVEE ,IVAL ,
6  ISPW ,ISPE ,IATD ,IRSP ,ISGT ,IDIE ,IDOI ,
7  ISGS ,NEXT ,IXPP ,ISGR ,ISL ,IES ,IWC ,
8  IEC ,ING ,IU ,IV ,IM ,ISI ,ST ,
9  INRG ,INSC ,INRP ,INCP ,IFXP ,IFXS ,
1  IFXT ,IFXE ,IFXA
COMMON/REGSCA/NSRPA,INSR
INSR = 0
NN = NNH
PDS = PDF
DO 10 I=1,3
X(I) = XXX(I)
10 C(I) = CCC(I)
CALL SZEUM(NMAX,5,1,NEMAX,HIISV),HIIXTR,HIINSO,HIISJN,
1  HIIEE,HIINPC,HIIVEE,HIIVAL,HIISPW,HIISPE,
2H(INS),H(IES),H(IIDV)
RETURN
END
SIFTC SZEUMI M94/2,XR7
CSZEUM=FIXED SOURCE EVALUATION FOR PROGRAM FASTER*J.M.JORDAN*MANL*1966*
SUBROUTINE SZEUM(I1,I2,I3,I4,ISV,XTR,NSG,SN,JSX,NPC,VEE,VAL,SPH,
1  ISPE,MS,ES,IV)
DIMENSION IDV(1)
DIMENSION Z(3),D(3),V(5),ROT(3,3)
DIMENSION ISV(1),XTR(3,1),NSG(1),JSN(1),JSX(1),NPC(5,1),ES(1),
1  VEE(1,2,3),VAL(1,2,3),SPH(4,1,3),SPE(1,2,3),MS(1)
EQUIVALENCE (Z(1),Z1),Z(2),Z2),Z(3),Z3),D(1),D1),D(2),D2),
1  D(3),D3),V(1),V1),V(2),V2),V(3),V3),V(4),V4),V(5),V5)
COMMON/DUMSZ/NN ,PCS ,X(3) ,C(3)
COMMON/POINTX/TOTAL,III ,MOD
COMMON/LIMITS/NSICRE,NERROR,NSMAX ,NMAX ,NRMAX ,NMAX ,NSTMAX,
1  NEMAX ,NVMAX ,NXPAX ,NXMAX,NEMOD ,NXSECT,NUNITO,
2  NUNITX,NMAX ,NMAX ,NORDER,NODOWN ,INELAS,NTRANS,

```

FAST1780
FAST1781
FAST1782
FAST1783
FAST1784
FAST1785
FAST1786
FAST1787
FAST1788
FAST1789
FAST1790
FAST1791
FAST1792
FAST1793
FAST1794
FAST1795
FAST1796
FAST1797
FAST1798
FAST1799
FAST1800
FAST1801
FAST1802
FAST1803
FAST1804
FAST1805
FAST1806
FAST1807
FAST1808
FAST1809
FAST1810
FAST1811
FAST1812
FAST1813
FAST1814
FAST1815
FAST1816
FAST1817
FAST1818
FAST1819
FAST1820
FAST1821
FAST1822
FAST1823
FAST1824
FAST1825
FAST1826

FAST1827
FAST1828
FAST1829
FAST1830
FAST1831
FAST1832
FAST1833
FAST1834
FAST1835
FAST1837
FAST1838
FAST1839
FAST1840
FAST1841
FAST1842
FAST1843
FAST1844
FAST1845
FAST1846
FAST1847
FAST1848
FAST1849
FAST1850
FAST1851
FAST1852
FAST1853
FAST1854
FAST1855
FAST1856
FAST1857

FAST1859
FAST1860
FAST1861
FAST1862
FAST1863
FAST1864
FAST1865
FAST1866
FAST1867
FAST1868
FAST1869
FAST1870

FAST1871
FAST1872
FAST1873

```

      SPH = 0.0
      GO TO 60
C SPHERICAL, COSINE OF POLAR ANGLE
      50 V3 = Z3/V1
      IF(INN.LE.0) GC TO 105
C ROTATION SETUP, POLAR
      CPH = V3
      SPH = SQRT(1.0 - CPH**2)
      R = V1*SPH
C ROTATION SETUP, AZIMUTHAL
      60 IFR,GT.0.0)GO TO 7C
      CTH = 1.0
      STH = 0.0
      GO TO 80
      70 CTH = Z1/R
      STH = Z2/R
C CALCULATE ROTATION MATRIX
      80 CALL ROTATE(CPH,SPH,CTH,STH,ROT)
C ROTATED DIRECTION COSINES
      90 I=1,3
      D(I) = 0.0
      DO 90 J=1,3
      90 D(I) = D(I) + ROT(I,J)*C(J)
C COSINE OF POLAR ANGLE, ANGULAR POF
      100 V5 = D3
C AZIMUTHAL ANGLE, ANGULAR POF
      V4 = 0.0
      IF(ABS(D1) + ABS(D2)).GT.0.0) V4 = ATAN2(D2,D1)
C ALL SOURCE VARIABLES DEFINED, SET ERROR INDICATOR, TOTAL POF
      JMIN = JS(N)
      JMAX = 0
      105 CONTINUE
      DO 130 I=1,5
C CHECK FOR DELTA FUNCTION, ASSUMED CORRECT
      IF(NPC(I),N).EQ.1)GO TO 130
C NOT DELTA FUNCTION, CHECK RANGE
      K = NPC(I,N)
      IF((V(I) - VEE(I,N))*(VEE(K,I,N) - V(I)).LT.0.0)GO TO 150
C INSIDE RANGE, FIND INTERVAL
      DO 110 J=2,K
      IF(VEE(J,I,N).GE.V(I))GO TO 120
      110 CONTINUE
C LINEAR INTERPOLATION
      120 K = J - 1
      PDS=PDS*(VAL(J,I,N)*(V(I)-VEE(K,I,N))+VAL(K,I,N)*(VEE(J,I,N)-V(I)))
      1 J/(VEE(J,I,N)-VEE(K,I,N))
      IF(INN.GT.0) GC TO 130
      130 CONTINUE
      IF(I.LT.3) GO TO 130
      PDS5 = PDS
      GO TO 150
      130 CONTINUE
C DEFINE NUMBER IN GROUP AND AVERAGE ENERGY
      JMAX = JS(N)
      DO 140 I=JMIN,JMAX
      WS(I) = PDS*SPW(I,N)
      140 ES(I) = SPE(I,N)
C REGULAR AND OUTSIDE-RANGE RETURNS
      150 RETURN
      END
      $B0FC KERNLE M94/2,XR7
      CKERNLE MEAN FREE PATH CALCULATION FOR PROGRAM FASTER* T.M. JORDAN*WANL*66
      SUBROUTINE KERNEL(MIN,MAX,ST,NRG)
      COMMON H(I)
      COMMON/LIMITS/NSSTORE,NEERROR,NSMAX,NAMAX,NRMAX,NBMAX,NTMAX,
      1 NEMAX,NYMAX,NZMAX,NXEMAX,NYEMAX,NZEMAX,NHMAX,NHSECT,NUNITO,
      2 NUNITX,NHMAX,NHMAX,NHORDER,NHODN,NINELAS,NTRANS,
      3 NNMAX,NGMAX,NFMAX,NVMOD,NCHMAX,NLMAX,NTMAX,
      4 NDMAX,NGMOD,NOMENT,NMODD,NPDMAX,NPDMOD,NSDMAX,
      5 NVDMAX,NVDMOD,NPOINT,MODEL,P,MODELQ,MODELU,MODELV,
      6 NPRINT,NUNITS,NUMBER,KALIDE
      COMMON/INDEXS/INTP,IAZ,ITSV,IRTL,IRHO,IXR,IELL,
      1 IELW,IAE,IBE,INSQ,INPC,IJSH,IJSX,
      2 IXTA,ITSV,IATN,IATW,IESB,ISSH,IDEH,
      3 INTG,IELF,IFGW,ITDS,ITDV,IFOR,IFDS,
      4 IVOL,ICOT,IXGT,IRSI,IALP,IVMD,IGIM,
      5 IAIM,IALH,IALH,IA,INS,IVEE,IVAL,
      6 ISW,ISPE,IATD,IRSP,ISGT,ILDE,ILDI,
      7 ISGS,NEXT,IXNP,ISGR,IMS,IES,INC,
      8 IEC,END,IU,IV,IW,ISI,IST,
      9 INRG,INSC,ISTP,INRP,INCP,IFXP,IFXS,
      1 IFX1,IFX2,IFX3
      CALL KERDUM(NHMOD,I,MI(XMP),MI(AE),MI(IBE),MI(IELL),MI(ELW),MI(IES),
      1 MI(IRHO),MI(ISSH),MI(INTL),MI(ISTG),MIN,MAX,ST,NRG)
      RETURN
      END
      $B0FC KERDML M94/2,XR7
      CKERDUM MEAN FREE PATH CALCULATION FOR PROGRAM FASTER* T.M. JORDAN*WANL*66
      SUBROUTINE KERDUM(I,L2,XMP,AE,BE,ELL,ELW,ES,RHD,SSH,MTL,SGT,MIN,
      1 MAX,ST,NRG)
      DIMENSION XMP(1),AE(1),BE(1),ELL(1),ELW(1),ES(1),NRG(1),MTL(1),
      1 ST(1),RHD(1),SGT(1),SSH(1)
      COMMON/OTHERS/RADIOUS,XCT(3),DELTA,DBDC(3),ATA,ATB,ATC,
      1 ATD,ATE,ATF,ATG,ATJ,ATK,ATL,ATM,ATN,ATO,ATP,ATQ,ATR,ATS,
      2 KMIN,KMAX,JBAR,NZERO

```

```

6      SPW ,LSPSE ,IATD ,LRSP ,LSGT ,LTIDE ,LTIDI , FAST20197
7      ISGS ,NEXT ,IXMP ,ISGR ,LVES ,LMC , FAST20198
8      IEC ,IND ,IU ,IV ,LW ,ISI ,IST , FAST20199
9      INRG ,INSC ,ISTP ,INRP ,INCP ,IFXP ,IFS , FAST20200
1     IFX1 ,IFXE ,IFXA , FAST20201
COMMON/REGGSA/INSRMAX,INSR
DIMENSION N(1),H(1)
EQUIVALENCE (N(1),H(1))
NN = NNN
PSI = AAA
POA = BBB
INSR = 0
IF(INSRMAX.LE.0) GO TO 10
DO 5 I=1,NSRMAX
IF(NN.EQ.N(I)) INSR = I
5 CONTINUE
10 CONTINUE
CALL SINDUM(NDOWN,NORDER,I,INELAS,NEMAX,H(ITLI),H(IRHO),H(TIDE),
1 H(ITDI), H(ISGS),H(IESB),H(ISSH),H(IDEN),
2 H(ITS),H(ITES),H(INCI),H(IECI),H(TELL))
RETURN
END
SUBFC SINDMI M94/2,XRT
CSINDUM=SINDMI SCATTERING.CALCULATION FOR PROGRAM FASTER*,M.JORDAN*ANDF
SUBROUTINE SINDUM(L1,L2,L3,L4,L5,MTL,RHO,IDE,TOT,SGS,ESB,SSH,
DECNVS,E5,MC,ECL)
DIMENSION MTI(1),RHO(1),TDE(1),L2,L3,JDCI(4,L3),SGSI(1
1 ,SSHI(1),DEN(1),H(1),ESI(1),ECI(1),ELL(1),KMX(1)
COMMON/DOMS/NHNN ,PSI ,POA
COMMON/Others/RADCT,DELTA,BDC(3),ATA ,ATB ,ATC ,
2 ATD ,AT ,BT ,BS ,BJMN ,JMAX ,
3 KMIN ,KNAX ,JBAR ,NZERO
COMMON/LIMITS/HSTORE,NERROR,MNAX ,NMAX ,NRMAX ,NHMAX ,NSTMAX ,
1 NMNX ,NGMAX ,NFMFX ,NVNMD ,NCMAX ,NLNAX ,NTMAX ,
2 NOXAX ,NGROD ,KOPENT,NODMO ,NPDMAX ,NPOMOD ,NEDMAX ,
3 NVOKAX ,NVONCD,NPCINT,MODELP,NPDQO,MODELU,MODELV ,
4 NPUNIT,NUNITS,NNUMBER,KALIDE
FST = 0.0
M = MTL(NNN)
JRMN = NEKAX + 1
JMAX = 0
DO 10 I=1,NEMAX
WS(I) = 0.0 --
10 ES(I) = 0.0
DO 20 J=1,OZ DO 20 JO

```

```

IF(NXSECT.NE.1)GO TO 220
100 PSIN = PDA
IF(M.EQ.0)GO TO 175
PLM2 = 0.0
PLM1 = 1.0
DO 139 L=1,NORDER
DO 120 K=1,NODOWN
IF(IDE(K,L,M).LE.0)GO TO 130
JMIN = KMIN
N = IDE(K,L,M)
KNOD = MINO(KMAX,NEMAX + 1 - K)
IF(KMIN.GT.KNOD)GO TO 130
I = KMIN + K - 2
DO 110 J=KMIN,KNOD
I = I + 1
110 WS(I) = WS(I) + WC(J)*SGS(J,N)*PSIN
JMAX = MAXO(JMAX,I)
120 CONTINUE
130 PLM2=(PSI*PLM1*FLOAT(2*L-1)-PLM2*FLOAT(L-1))/FLOAT(L)
PLM2 = PLM1
PLM1 = PLM2
PSIN = PDA*PLM2
139 CONTINUE
KNOD = MINO(KMAX,INELAS)
IF(KMOD.LT.KMIN) GO TO 160
DO 150 J=KMIN,KNOD
K = IO(I,J,N)
IF(K.EQ.0) GO TO 160
N = K/1000
MAX = MINO(NEMOD-J,K-1000*N)
I = J - 1
DO 140 K=1,MAX
I = I + 1
140 WS(I) = WS(I) + MC(J)*SGS(K,N)*PDA
JMAX = MAXO(J,JMAX)
150 CONTINUE
160 DO 170 I=JMIN,JMAX
170 ES(I) = ES(I)*WS(I)
FST = DER(N)
175 FST = 4.0*PSI*PDA*(FST + RHO(N))
IF(FST.LE.0.0)GO TO 900
RAT = PSI**2
MIN = KMIN
DO 200 J=KMIN,KMAX
E = RAT*EC(I)
DO 180 I=MIN,NEMAX
IF(E-GE,ELL(I+1))GO TO 190

180 CONTINUE
GO TO 210
190 MIN = I
JMIN = MINO(JMIN,I)
WSC = FST*SSH(J)*WC(J)
WS(I) = WS(I) + WSC
200 ES(I) = ES(I) + WSCPE
210 JMAX = MAXO(JMIN,JMAX)
GO TO 900
COMPTON SCATTERING
220 MIN = KMIN
PSIN = 1.0 - PSI
PSIS = PSI**2 - 1.0
GST = RHO(IN)*0.49875
IF(M.GT.0)GST = GST + DER(M)
GST = PDA*GST
DO 250 J=KMIN,KMAX
RAT = 0.511/(0.511 + PSIN*EC(J))
E = RAT*EC(J)
DO 230 I=MIN,NEMAX
IF(E-GE,ELL(I+1))GO TO 240
230 CONTINUE
GO TO 260
240 MIN = I
JMIN = MINO(JMIN,I)
FST = WC(J)*GST*RAT**2*(RAT + 1.0/RAT + PSIS)
WS(I) = WS(I) + FST
ES(I) = ES(I) + FST*E
250 CONTINUE
260 JMAX = MIN
900 IF(JMIN.GT.JMAX)GO TO 920
DO 910 I=JMIN,JMAX
IF(WS(I).NE.0.0) ES(I) = ES(I)/WS(I)
910 CONTINUE
920 RETURN
END
*BTFTC TRACE M94/2,XR7
CTRACER*PARTIAL PATH LENGTHS FOR PROGRAM FASTER*J.H.JORDAN*WANL*1966*
SUBROUTINE PATH (NNN,SSS,XXX,CCC,MMN,ST,NRG,NSC)
COMMON H(I)
COMMON/DUNTRX/NN ,STW ,NST ,X(9) ,C(9) ,CX(9)
COMMON/LIMITS/NSTORE,NEROR,NMAX ,NMAX ,NRMAX ,NBMAX ,NSTMAX,
1 NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD,
2 NUNITX,NIMAX ,NMWAX ,NCRDER,NDOWN ,INELAS,NTRANS,
3 NRMAX ,NRMAX ,NFMAX ,NVNOD ,NCMAX ,NLMAX ,NTMAX ,
4 NDMAX ,NGMOD ,NDMENT,NMOD ,NPDMAX,NPCMOD,NSMAX,
5 NVDMAX,NVDMOD,NPCINT,NODELP,NODEQL,MODELU,MODELV,

```

FAST2060
FAST2061
FAST2062
FAST2063
FAST2064
FAST2065
FAST2066
FAST2067
FAST2068
FAST2069
FAST2070
FAST2071
FAST2072
FAST2073
FAST2074
FAST2075
FAST2076
FAST2077
FAST2078
FAST2079
FAST2080
FAST2081
FAST2082
FAST2083
FAST2084
FAST2085
FAST2086
FAST2087
FAST2088
FAST2089
FAST2090
FAST2091
FAST2092
FAST2093
FAST2094
FAST2095
FAST2096
FAST2097
FAST2098
FAST2099
FAST2100
FAST2101
FAST2102
FAST2103
FAST2104
FAST2105
FAST2106

```

6      NPRINT,NUNITS,NUMBER,KALIDE          FAST2154
COMMON/INDEXS/INTP ,IAZ ,IISV ,IKTL ,IRHD ,IXR ,IELL          FAST2155
1      IELW ,IAE ,IBE ,INSG ,INPC ,IJSH ,IJSK          FAST2156
2      IATR ,ISUV ,IATN ,IATW ,IESB ,ISSH ,IDEN          FAST2157
3      INTG ,IELF ,IFGW ,ITDS ,IDV ,IIDR ,IIDS          FAST2158
4      IVOL ,ICDT ,IXDT ,IRSI ,IALP ,IVMD ,IGTM          FAST2159
5      IAIM ,IALM ,IALH ,IA ,INS ,IVER ,IVAL          FAST2160
6      ISPW ,ISPE ,IATO ,IRSP ,ISOT ,IIOE ,IIOI          FAST2161
7      ISGS ,NEXT ,IXMP ,ISGR ,IWS ,IES ,IWC          FAST2162
8      IEC ,IND ,IU ,IV ,IWH ,ISI ,IST          FAST2163
9      INRG ,INSG ,ISTP ,INRP ,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(NMAX,NMAX,I,HINTP,HIAZ,HIAI,HINSI,HENDI,          FAST2172
1     HITU,HIVI,HIMI,HISI,ST,NRG,NSC)          FAST2173
      MM = NST          FAST2174
      RETURN          FAST2175
      ENDO          FAST2176
$IBFFC TRADUM M94/2,XR7          FAST2177
      CTADUMPARTIAL PATH LENGTHS FOR PROGRAM FASTER* M.JORDAN*MANL*1966*          FAST2178
      SUBROUTINE TRADUM(L1,L2,L3,NTP,AZ,AS,NS,ND,U,V,W,SI,ST,NRG,NSC)          FAST2179
      DIMENSION NTP(1),AZ(1),AL(1,3),NST(2,3),ST(1),NRG(1),NSC(1),          FAST2180
1     ND(1),U(1),V(1),W(1),ST(2,1)          FAST2181
      COMMON/DUNTRA/NN ,STM ,NST ,X(9) ,C(9) ,CX(9)          FAST2182
      COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NMAX ,NRMAX ,NDMAX ,NSTMAX,          FAST2183
1     NEMAX ,NMWAX ,NEMAX ,NEMAX ,NEMOD ,NMSCT,NUNITO,          FAST2184
2     NUNITX,NIMAX ,NMWAX ,NORDER,NDOWN ,INELAS,NTRANS,          FAST2185
3     NRMAX ,NGMAX ,NFMWAX ,NVWOD ,NCMAX ,NLMAX ,NTMAX,          FAST2186
4     NDMAX ,NHMOD ,NMOENT ,NHMOD ,NPDMAX ,NPDOD ,NSDMAX,          FAST2187
5     NVDMAX ,NVDMOD ,NPOINT ,MODEL ,MODELO ,MODELU ,MODELV,          FAST2188
6     NPRINT,NUNITS,NUMBER,KALIDE          FAST2189
      DO 110 I=1,3          FAST2190
        J = 1 + I - 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,NMAX          FAST2218
      IF(NS(I,I).EQ.0)GO TO 390          FAST2219
      IF(ABS(NSP - NS(I,I/1000)).EQ.1000)GO TO 190          FAST2220
180    CONTINUE          FAST2221
      GO TO 390          FAST2222
190    DO 360 J=1,NMAX          FAST2223
      IF(NS(J,I).EQ.0)GO TO 350          FAST2224
      KK = NST(J,I)/1000000          FAST2225
      K = NS(J,I)/1000 - 1000*KK          FAST2226
      IF((K.GT.1).AND.(K.EQ.L1))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)*AL(K)          FAST2236
      V(K) = V(K) + C(L)*AL(K)          FAST2237
      IF(L.GT.3)W(K) = W(K) + CX(L)*AL(K)          FAST2238
200    CONTINUE          FAST2239
210    IF(M.LE.1)GO TO 220          FAST2240
220    IF(FLD(K)*U(K) + STT*(V(K) + STT*(W(K))) 340,34C,390          FAST2241
230    IF(ND(K) - 21 230,290,340          FAST2242
230    ND(K) = 2          FAST2243
      IF(W(K)) 250,240,240          FAST2244
240    IF(V(K).EQ.0)GO TO 310          FAST2245
      SI(L,K) = -U(K)/V(K)          FAST2246
      IF(SI(L,K).LT.0)GO TO 310          FAST2247

```

```

      S(I2,K) = S(I1,K)
      GO TO 290
250 L = 2
      GO TO 270
260 L = 1
270 E = -0.5*W(K)/W(K)
      F = U(K)/W(K)
      H = E**2 - F
      IF(F.LE.0.01GO TO 280
      IF(E.LE.0.01GO TO 310
      IF(H.LE.0.01GO TO 310
280 H = SQRT(H)
      S(I1,K) = E - H
      L = L + 1 - 2*(L/2)
      S(I1,K) = E + H
290 L = 1
      IF(KK.GT.01 = 2
      SB = S(I1,K) - STT
      IF(SB) 300,320,330
300 L = L + 1 - 2*(L/2)
      IF(S(I1,K).GT.STT)GO TO 340
310 NDK(K) = 3
      GO TO 340
320 IF(FLOAT(KK)*(V(K) + 2.0*STT*W(K)).LT.0.01GO TO 340
330 IF(SB.OE.SPIGO TO 340
      SP = SB
      JJ = J
      LL = K
340 CONTINUE
350 IF(M.LE.1)GO TO 360
      II = I
      NS(JJ,II) = 1000*NSP + I
      GO TO 410
360 IF(JJ.EQ.01GO TO 415
370 NRG(N) = I
      IF(STN.GT.(STT + SP))GO TO 380
      NSC(N) = 0
      ST(N) = STN - STT
      GO TO 420
380 NSC(N) = LL
      ST(N) = SP
      STT = STT + SP
      IF(INTP(LL,LE,3)ND(LL) = 3
      NSP = NS(JJ,II)/1000
      NXT = NS(JJ,II) - 1000*NSP
      NXXX = 0
      IF(NXT.LE.NRMXX) GO TO 385

```

```

      NXXX = NXT - NRMXX
      IF(NXXX.GT.10) GO TO 420
      NXT = 0
385 IP = I
390 CONTINUE
400 CONTINUE
      NS(JJ,IP) = 1000*NSP + MIN(999,NRMXX+NXXX+1)
      GO TO 420
410 CONTINUE
      N = NSTMAX + 1
415 N = N - 1
420 NST = N
430 RETURN
      END
*18FTC N94/2,XR7
CNDROM1=SURFACE NORMAL CALCULATION FOR PROGRAM FASTER*1.M.JORDAN*MANL*66F
      SUBROUTINE NORMAL(NNN,SSS,XXX,CCC,DDD)
      COMMON H(I)
      COMMON/LIMITS/HSTORE,NERROR,NMAX,NAMAX,NRMXX,NBMAX,NSTMAX,
1 NEMAX,NVMAX,NXMAX,NEMAX,NEMOD,NXSECT,NUNITD,
2 NUNITX,NIMAX,NMPAX,NORDER,NODWN,INELAS,NTRANS,
3 NNMXX,NGMXX,NFMAX,NVMD,NCMXX,NLMXX,NIMAX,
4 NDMAX,NGMOD,NPMENT,NOMOD,NPDMAX,NPDMOD,NSDMAX,
5 NVDMAX,NVDMCD,NPOINT,MODELP,MODELO,MODELU,MODELV,
6 NPRINT,NUNITS,NURBER,KALIDE
      COMMON/INDEXS/INTP,IAZ,II5V,INTL,IRHO,IXR,IELL,
1 IELW,IAE,IEE,INSG,IMPC,IJSN,IJSX,
2 IXTX,ISUV,IATN,IATW,IESB,ISSH,IOEN,
3 INTG,IELF,IFGW,ITOS,IIOV,IIDR,IIDS,
4 IVOL,ICOT,IXCT,IRSI,IALP,IVMD,IGIM,
5 IATX,IALM,IALW,IA,IEIS,IVEE,IWAL,
6 ISPW,ISPE,IATO,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
      CALL NORUDM(NAMAX,1,H,INTP),H(IA),NNN,SSS,XXX,CCC,DDD)
      RETURN
      END
*18FTC N94/2,XR7
CNDROM1=SURFACE NORMAL CALCULATION FOR PROGRAM FASTER*1.M.JORDAN*MANL*66F
      SUBROUTINE NORUDM(L1,L2,NTP,A,NNN,STT,XP,CP,C)
      DIMENSION NTP(1),X(L1),L21,XP(3),CP(3),X(3),C(3)
      DO 10 I=1,3
10 X(I) = XP(I) + STT*CP(I)
1 = NNN
      FST = 0.0

```

FAST2249
FAST2250
FAST2251
FAST2252
FAST2253
FAST2254
FAST2255
FAST2256
FAST2257
FAST2258
FAST2259
FAST2260
FAST2261
FAST2262
FAST2263
FAST2264
FAST2265
FAST2266
FAST2267
FAST2268
FAST2269
FAST2270
FAST2271
FAST2272
FAST2273
FAST2274
FAST2275
FAST2276
FAST2277
FAST2278
FAST2279
FAST2280
FAST2281
FAST2282
FAST2283
FAST2284
FAST2285
FAST2286
FAST2287
FAST2288
FAST2289
FAST2290
FAST2291
FAST2292
FAST2293
FAST2294

FAST2295
FAST2296
FAST2297
FAST2298
FAST2299
FAST2300
FAST2301
FAST2302
FAST2303
FAST2304
FAST2305
FAST2306
FAST2307
FAST2308
FAST2309
FAST2310
FAST2311
FAST2312
FAST2313
FAST2314
FAST2315
FAST2316
FAST2317
FAST2318
FAST2319
FAST2320
FAST2321
FAST2322
FAST2323
FAST2324
FAST2325
FAST2326
FAST2327
FAST2328
FAST2329
FAST2330
FAST2331
FAST2332
FAST2333
FAST2334
FAST2335
FAST2336
FAST2337
FAST2338
FAST2339
FAST2340
FAST2341


```

1SSH,SGT,XMP,GIM,P,T,AIM)
DIMENSION WS(11),ES(1),NTG(1),FXT(L2,L1),FXP(L2,L1),FXE(L2,L1),
1FXA(L4,L1),RHO(1),AE(1),BSH(1),SGT(L3,L1),XMP(1),GIM(1),
2P(1),T(1),AIM(1)
COMMON/DUMOST/1,M,FST,ANG,STV
COMMON/OTHERS/RADIUS,XCT(3),DELTA,80C(3),ATA,ATB,ATC,
1ATO,AT,AT,AT,AS,BS,JMIN,JMAX,
2KMIN,KMAX,JBAR,NZERO
COMMON/FLUXES/KKK,NTALLY,ERROR,TOTALN,TOTALE,RHON,SNORM,TOTFAS
COMMON/LIMITS/NSTORE,NERROR,NSMAX,NAMAX,NRMAX,NBMAX,NSTMAX,
1NEMAX,NVMAX,NXMAX,NXEMAX,NEMOD,NXSECT,NUNITD,
2NUNITX,NIMAX,NHMAX,NORDR,NODMN,INELAS,NTRANS,
3NNMAX,NGMAX,NFMAX,NVMO,NCGMAX,NLMAX,NTHAX,
4NDMAX,NGMOD,NMOMENT,NMODD,NPDMAX,NPOMOD,NSDMAX,
5NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,
6NPRINT,NUNITS,NUMBER,KALIDE
COMMON/POINTX/TOTAL,11,HOP
COMMON/REGSCA/NSRMAX,INSR
C ANGULAR MOMENTS
1F(NLMAX,LE,0)GO TO 20
P(1) = ANG
P(2) = 1.5*ANG*ANG-0.5
1F(NLMAX,LE,2)GO TO 20
DO 10 L=3,NLMAX
20 P(L) = (FLOAT(2*L-1)*ANG*P(L-1) - FLOAT(L-1)*P(L-2))/FLOAT(L)
1F(NLMAX,LE,0)GO TO 30
1FIM,GE,0)GO TO 30
T(1) = TOT/SHORN
1F(NMAX,EQ,1)GO TO 30
DO 25 L=2,NMAX
25 T(L) = T(1)*T(L-1)
30 DO 80 K=JMIN,JMAX
1FWS(K),EQ,0
GST = FST*WS(K)*EXP(-XMP(K)) JGO TO 80
1F(N,LT,0)GO TO 40
SIG = RHON *TAE(K)*SSH(K) + BE(K)*SSH(K+1)
1F(K,GT,0)SIG = SIG + IAE(K)*SGT(K,M) + BE(K)*SGT(K+1,M)
ATN = EXP(-SIG*STV)
AST = 11.0 - ATN/SIG
GST = GST*AST
XMP(K) = XMP(K) + SIG*STV
40 1F(GST,EQ,0)GO TO 80
J = NTG(K)
FXT(J,1) = FXT(J,1) + GST
FXP(J,1) = FXP(J,1) + GST
FXT(J,1) = FXT(J,1) + GST*ES(K)
HST = GST*XMP(K)*GIM(K)*AIM(K)
FAST2436
FAST2437
FAST2438
FAST2439
FAST2440
FAST2441
FAST2442
FAST2443
FAST2444
FAST2445
FAST2446
FAST2447
FAST2448
FAST2449
FAST2450
FAST2451
FAST2452
FAST2453
FAST2454
FAST2455
FAST2456
FAST2457
FAST2458
FAST2459
FAST2460
FAST2461
FAST2462
FAST2463
FAST2464
FAST2465
FAST2466
FAST2467
FAST2468
FAST2469
FAST2470
FAST2471
FAST2472
FAST2473
FAST2474
FAST2475
FAST2476
FAST2477
FAST2478
FAST2479
FAST2480
FAST2481
FAST2482

```

```

HST = ABS(HST)
1F(ABS(GST),GT,(ERROR*FXP(J,1))) NTALLY = NTALLY + 1
TOTALN = TOTALN + HST
TOTALE = TOTALE + HST*ES(K)
FXA(1,J,1) = FXA(1,J,1) + GST
KK = MOD + 1
1F(KK,GT,1) FXA(KK,J,1) = FXA(KK,J,1) + GST
LL = NVMOD + 1
KK = LL + INSR
1F(INSR,GT,0) FXA(KK,J,1) = FXA(KK,J,1) + GST
LL = LL + NSRMAX
KK = LL + KKK
1F(KKK,LE,NCGMAX) FXA(KK,J,1) = FXA(KK,J,1) + GST
KK = LL + NCGMAX
1F(NLMAX,LE,0)GO TO 60
DO 50 L=1,NLMAX
KK = KK + 1
50 FXA(KK,J,1) = FXA(KK,J,1) + GST*P(L)
60 1F(NMAX,LE,0)GO TO 80
1F(N,LT,0)GO TO 68
AST = SIG - STV*ATN/11.0 - ATN)
T(1) = (TOT + AST)/SHORN
1F(NMAX,GO,1)GO TO 68
DO 63 L=2,NMAX
63 T(L) = T(1)*T(L-1)
68 CONTINUE
DO TO L=1,NMAX
KK = KK + 1
70 FXA(KK,J,1) = FXA(KK,J,1) + GST*T(L)
80 CONTINUE
RETURN
END
SUBROUTINE ANSWR N94/2,XR7
CAUSHER*FLUX PRISTOUT FOR PROGRAM FASTER
SUBROUTINE ANSWR(NNN)
COMMON H(1)
COMMON/LIMITS/NSTORE,NERROR,NSMAX,NAMAX,NRMAX,NBMAX,NSTMAX,
1NEMAX,NVMAX,NXMAX,NXEMAX,NEMOD,NXSECT,NUNITD,
2NUNITX,NIMAX,NHMAX,NORDR,NODMN,INELAS,NTRANS,
3NNMAX,NGMAX,NFMAX,NVMO,NCGMAX,NLMAX,NTHAX,
4NDMAX,NGMOD,NMOMENT,NMODD,NPDMAX,NPOMOD,NSDMAX,
5NVDMAX,NVDMOD,NPOINT,MODELP,MODELQ,MODELU,MODELV,
6NPRINT,NUNITS,NUMBER,KALIDE
COMMON/INDEXS/INTP,IAZ,ITSV,IMTL,IRHO,IXR,IELL,
1IELM,IAE,IOE,IKSG,INPC,IJSN,IJSA,
2IATN,ISUV,IATN,IESB,ISSH,IOEN,
3INTG,IELF,IFEW,ITOS,IOV,ITOR,IIDS,
FAST2483
FAST2484
FAST2485
FAST2486
FAST2487
FAST2488
FAST2489
FAST2490
FAST2491
FAST2492
FAST2493
FAST2494
FAST2495
FAST2496
FAST2497
FAST2498
FAST2499
FAST2500
FAST2501
FAST2502
FAST2503
FAST2504
FAST2505
FAST2506
FAST2507
FAST2508
FAST2509
FAST2510
FAST2511
FAST2512
FAST2513
FAST2514
FAST2515
FAST2516
FAST2517
FAST2518
FAST2519
FAST2520
FAST2521
FAST2522
FAST2523
FAST2524
FAST2525
FAST2526
FAST2527
FAST2528
FAST2529

```


240

```

OUT3 = FXT(J,I)/XN
IF(OUT3.EQ.0.01G0 TO 100
OUT4 = FAE(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.0.01OUT1) = SQRT(FST)/OUT3
100 IF(LABEL(1).GE.0)WRITE(M2,2010)I,OUT
2010 FORMAT(1X,5HGROUP,14,3H...,1PE11.4,7E12.4)
AE(L) = (OUT2 - ELF(J+1))/FGW(J)
BE(L) = (ELF(J) - OUT2)/FGW(J)
FLX(J) = OUT3
110 ERR(J) = OUT1
IF(NFMAX.EQ.01G0 TO 170
DO 160 J=1,NFMAX,8
MAX = MIN(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)FAST2624
2020 FORMAT(1X,24(1H*),34HNUMBER FLUX RESPONSES FOR DETECTOR,14,6H AFTERFAST2625
1R,16,8H PACKETS,25(1H*)/12X,24A4)FAST2626
LINES = LINES - 1FAST2627
DO 120 K=1,MODFAST2628
OUS(K) = 0.0FAST2629
ENH(K) = 0.0FAST2630
120 ENH(K) = 0.0FAST2631
DO 140 K=1,NGMAXFAST2632
L = 0FAST2633
DO 130 M=J,MAXFAST2634
L = L + 1FAST2635
FST = FLX(K)*(AE(K)*RSP(K,M) + BE(K)*RSP(K+1,M))FAST2636
OUT(L) = FSTFAST2637
OUS(L) = OUS(L) + FSTFAST2638
FST = FST*ERR(K)FAST2639
ENH(L) = ENH(L) + FST**2FAST2640
130 ENH(L) = ENH(L) + FSTFAST2641
IF(LABEL(1).GE.0)WRITE(M2,2010)K,(OUT(L),L=1,MOD)FAST2642
140 CONTINUEFAST2643
IF(LABEL(1).GE.0)WRITE(M2,2030)(OUS(L),L=1,MOD)FAST2644
2030 FORMAT(1X,6HTOTALS,6(1H*),1PE11.4,7E12.4)FAST2645
DO 150 L=1,MODFAST2646
OUT(L) = 0.0FAST2647
FST = OUS(L)FAST2648
OUS(L) = 0.0FAST2649
IF(FST.EQ.0.01G0 TO 150FAST2650
OUT(L) = SQRT(ENH(L))/FSTFAST2651
FAST2652
FAST2653
FAST2654
FAST2655
FAST2656
FAST2657
FAST2658
FAST2659
FAST2660
FAST2661
FAST2662
FAST2663
FAST2664
FAST2665
FAST2666
FAST2667
FAST2668
FAST2669
FAST2670

```

241

```

OUS(L) = ENH(L)/FST
150 CONTINUE
IF(LABEL(1).GE.0)WRITE(M2,2035)IOUT(L),L=1,MOD)
2035 FORMAT(1X,12HMIN ERROR...,1PE11.4,7E12.4)
IF(LABEL(1).GE.0)WRITE(M2,2040)IOUT(L),L=1,MOD)
2040 FORMAT(1X,12HMAX ERROR...,1PE11.4,7E12.4)
160 CONTINUE
170 IF(NMOD.EQ.01G0 TO 240
DO 230 J=1,NMOD,8
MAX = MIN(J+7,NMOD)
MOD = MAX - J + 1
IF(LABEL(3).GE.0)WRITE(M2,2050)IOUT,NPF,((TLE(K,L),K=1,2),L=1,MOD)FAST2671
1=J,MAX)FAST2672
2050 FORMAT(1X,25(1H*),32HNUMBER FLUX MOMENTS FOR DETECTOR,14,6H AFTER,FAST2673
116,8H PACKETS,26(1H*)/12X,8(1X,2A4,13))FAST2674
LINES = LINES - 1FAST2675
DO 190 K=1,NGMAXFAST2676
L = 0FAST2677
DO 180 M=J,MAXFAST2678
L = L + 1FAST2679
180 OUT(L) = FXA(H,K,I)/XNFAST2680
IF(L.EQ.1) OUT1 = XT*OUT1FAST2681
IF(LABEL(1).GE.0)WRITE(M2,2010)K,(OUT(L),L=1,MOD)FAST2682
190 CONTINUEFAST2683
IF(NFMAX.EQ.01G0 TO 230FAST2684
DO 220 K=1,NFMAXFAST2685
DO 200 L=1,MODFAST2686
200 OUT(L) = 0.0FAST2687
DO 210 L=1,NGMAXFAST2688
M = 0FAST2689
DO 210 N=J,MAXFAST2690
M = M + 1FAST2691
210 OUT(M) = OUT(M) + FXA(H,K,I)*(AE(L)*RSP(L,K)+BE(L)*RSP(L+1,K))/XNFAST2692
IF(L.EQ.1) OUT1 = XT*OUT1FAST2693
IF(LABEL(1).GE.0)WRITE(M2,2060)IOUT,((TDS(L,K),L=1,3),IOUT(L),L=1,MOD)FAST2694
2060 FORMAT(1X,3A4,1PE11.4,7E12.4)FAST2695
220 CONTINUEFAST2696
230 CONTINUEFAST2697
240 CONTINUEFAST2698
DO 250 I=1,NMODFAST2699
DO 250 J=1,NGMAXFAST2700
250 FXA(I,J,I) = 0.0FAST2701
IF(NNN.LT.NPRINT)GO TO 900FAST2702
IF(NPOINT.EQ.01)GO TO 300FAST2703
IF(NPORUN.LT.NPRAX)GO TO 870FAST2704
300 IF(LABEL(1).GE.0)WRITE(M2,3000)FAST2705
3000 FORMAT(1X,22(1H*),63HBOUNDARY SEARCH PARAMETERS, (SURFACE,MOST PROF)FAST2717

```

FAST2718
FAST2719
FAST2720
FAST2721
FAST2722
FAST2723
FAST2724
FAST2725
FAST2726
FAST2727
FAST2728
FAST2729
FAST2730
FAST2731
FAST2732
FAST2733
FAST2734
FAST2735
FAST2736
FAST2737
FAST2738
FAST2739
FAST2740
FAST2741
FAST2742
FAST2743
FAST2744
FAST2745
FAST2746
FAST2747
FAST2748
FAST2749
FAST2750
FAST2751
FAST2752
FAST2753
FAST2754
FAST2755
FAST2756
FAST2757
FAST2758
FAST2759
FAST2760
FAST2761
FAST2762
FAST2763
FAST2764

FAST2765
FAST2766
FAST2767
FAST2768
FAST2769
FAST2770
FAST2771
FAST2772
FAST2773
FAST2774
FAST2775
FAST2776
FAST2777
FAST2778
FAST2779
FAST2780
FAST2781
FAST2782
FAST2783
FAST2784
FAST2785
FAST2786
FAST2787
FAST2788
FAST2789
FAST2790
FAST2791
FAST2792
FAST2793
FAST2794
FAST2795
FAST2796
FAST2797
FAST2798
FAST2799
FAST2800
FAST2801
FAST2802
FAST2803
FAST2804
FAST2805
FAST2806
FAST2807
FAST2808
FAST2809
FAST2810
FAST2811

```

      [FINSG(N) - 1] 60,80,90
60 DO 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)**ASG(N)
110 DO I20 I=1,3
120 X(I) = X(I) + XTR(I,N)
   NN = LOCATE(NGS,X)
   RETURN
END
110FTC SPHER M94/2,XR7
CSPHERE=INITIAL SOURCE VECTOR FROM SPHERE FOR PROGRAM FASTER*Y.P.JORDAN
FUNCTION SPHERE(YYY,AAA,ZZZ,HMM,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 ,NMAX ,NRMAX ,NBMAX ,NSTMAX,
1    NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,
2    NUNITX,NIMAX ,NMMAX ,NORDER,NODWN ,INELAS,NTRANS,
3    NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
4    NDMAX ,NGMOD ,NPOENT,NMOD ,NPOMAX ,NPOMOD ,NSGMAX,
5    NVDMAX ,NVDMOD ,NPCINT ,MDELP ,MDELO ,MODELU ,MODELV,
6    NPRINT,NUNITS,NUMBER,KALIDE
  COMMON/INDEXS/INTP ,IAZ ,IISV ,IMTL ,IRHO ,IXR ,IELL ,
1    IELM ,IAE ,IBE ,INSG ,INPC ,IJSN ,IJSX ,
2    IXTX ,ISUV ,IATN ,IATW ,IESB ,ISSH ,IDEN ,
3    INTG ,IELF ,IFGW ,ITDS ,ITDV ,ITDR ,ITDS ,
4    ITVL ,ICDT ,IACI ,IRSI ,IALP ,IVMD ,ICGR ,
5    IATM ,IALM ,IALH ,IA ,IMS ,IVEE ,IVAL ,
6    ISPW ,ISPE ,IATD ,IRSP ,ISGT ,ITDE ,ITDI ,
7    ISGS ,NEXT ,IXMP ,ISGR ,IWS ,IES ,IWC ,
8    IEC ,IND ,IDU ,IW ,IN ,IS1 ,IST ,
9    INRG ,INSC ,ISTP ,INRP ,INCP ,IFXP ,IFXS ,
1   IFXT ,IFXE ,IFXA
  DO I0 I=1,3
10 X(I) = YYY(I)
  CALL SPHDUM(NEMOD,L,I(1)SV,I(1)SU,I(1)RH,I(1)SSH,I(1)MTL,
1    I(1)SGT,I(1)IND,I(1)IV,I(1)IS,I(1)ST,NRG,NSC)
  HMM = NST
  AAA = STC
FAST2812
FAST2813
FAST2814
FAST2815
FAST2816
FAST2817
FAST2818
FAST2819
FAST2820
FAST2821
FAST2822
FAST2823
FAST2824
FAST2825
FAST2826
FAST2827
FAST2828
FAST2829
FAST2830
FAST2831
FAST2832
FAST2833
FAST2834
FAST2835
FAST2836
FAST2837
FAST2838
FAST2839
FAST2840
FAST2841
FAST2842
FAST2843
FAST2844
FAST2845
FAST2846
FAST2847
FAST2848
FAST2849
FAST2850
FAST2851
FAST2852
FAST2853
FAST2854
FAST2855
FAST2856
FAST2857
FAST2858

```

244

```

      DO I0 I=1,3
20 ZZZ(I) = C(I)
  SPHERE = PDT
  RETURN
END
110FTC SPHDH M94/2,XR7
CSPHDUM=INITIAL SOURCE VECTOR FROM SPHERE FOR PROGRAM FASTER*Y.P.JORDAN
SUBROUTINE SPHDUM(L,L2,ISV,SUV,RHO,SSH,MTL,SGT,XM,TRA,TRB,ST,NRG,
1    INSC)
  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 RCT(1),C(3)
  COMMON/DUMSPH/NST ,PDT ,STT ,X(3) ,C(3)
  COMMON/LIMITS/NSTORE,NERROR,NSMAX ,NMAX ,NRMAX ,NBMAX ,NSTMAX,
1    NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITO,
2    NUNITX,NIMAX ,NMMAX ,NORDER,NODWN ,INELAS,NTRANS,
3    NNMAX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NTMAX ,
4    NDMAX ,NGMOD ,NPOENT,NMOD ,NPOMAX ,NPOMOD ,NSGMAX,
5    NVDMAX ,NVDMOD ,NPCINT ,MDELP ,MDELO ,MODELU ,MODELV,
6    NPRINT,NUNITS,NUMBER,KALIDE
  COMMON/POINTX/NTOTAL,III ,NOP
  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/SPASH/PSORS
  DATA P1/3,141593/
  COMMON/LLSREG/LLREG
  DIMENSION NSC(1)
C NEW METHOD, SETUP SPHERICAL VOLUME
130 STM = VECTOR(X,XCT,C)
  CALL ROTAT(C,ROT)
  NN = LOCATE(NGS,X)
  PHMIN = -1.0
  FST = STM**2 - RADIUS**2
  IF(FST.GE.0.0)PHMIN = SQRT(FST)/STM
  STM = STM + RADIUS
  ATX = -ATA/(1.0 - PHMIN)
  DO I65 NTRY=1,100
    NTOTAL = NTOTAL + 1
    PDT = SAMPLE(PHMIN,1.0,1.0,ATX,DE3)**SAMPL,ET=PI,0.0,ATB,THE)
    SNP = SQRT(1.0 - D1(3)**2)
    D(1) = SNP*COS(THET)
    D(2) = SNP*SIN(THET)
  DO I50 I=1,3
    C(I) = 0.0
FAST2859
FAST2860
FAST2861
FAST2862
FAST2863
FAST2864
FAST2865
FAST2866
FAST2867
FAST2868
FAST2869
FAST2870
FAST2871
FAST2872
FAST2873
FAST2874
FAST2875
FAST2876
FAST2877
FAST2878
FAST2879
FAST2880
FAST2881
FAST2882
FAST2883
FAST2884
FAST2885
FAST2886
FAST2887
FAST2888
FAST2889
FAST2890
FAST2891
FAST2892
FAST2893
FAST2894
FAST2895
FAST2896
FAST2897
FAST2898
FAST2899
FAST2900
FAST2901
FAST2902
FAST2903
FAST2904
FAST2905

```

245

```

      DO 150 J=1,3
150  C(I) = C(I) + D(J)*RCT(J,I)
      CALL PATH(INN,STM,X ,C,NST,ST,NRG,NSC)
      STT = 0.0
      DO 160 I=1,NST
        J = NRG(I)
        IF (ISV(J).GT.0)GO TO 170
160  STT = STT + ST(I)
165  CONTINUE
      NST = 0
      GO TO 300
170  NTT = 1
      PTOT = 0.0
      XMPT = 0.0
      STX = STT
      DO 220 I=NTT,NST
        J = NRG(I)
        K = ISV(J)
        L = MTL(J)
        SGR = ATC*RHDI(J)*SSH(JZERCI)
        IF(L.GT.0)SGR = SGR + ATC*SGT(JZERO,L)
        XMPP = SGR*ST(I)
        IF(K.GT.0)GO TO 180
        XM(I) = 0.0
        GO TO 210
180  CALL ARROW(X,STX + 0.01*ST(I),C,XS)
        LLREG = K
        CALL SZERC(0,1.0,XS,C)
        POB = PSORS
        CALL ARROW (X,STX + 0.5*ST(I),C,XS)
        CALL SZERC(0,1.0,XS,C)
        PDM = PSOMS
        TRAF(I) = PDR*SUV(K)*EXP(-XMPT)
        ARGARG = 0.0
        IF((POB*PDM).GT.0.0) ARGARG = ALDG(PCB/PDM)
        IF(ST(I).GT.C.0) SGR = SGR + 2.0*ARGARG/ST(I)
        TRB(I) = SGR
        IF(SGR.LE.0.0)GO TO 190
        XM(I) = ST(I)*TRAF(I)
        GO TO 200
190  XM(I) = TRAF(I)*(1.0 - EXP(-SGR*ST(I)))/SGR
200  PTDI = PTOT + XM(I)
210  XMPT = XMPT + XMPP
220  STX = STX + ST(I)
      PDS = PTOT*PANH(INMP)
      DO 230 I=NTT,NST
        IF(XM(I).GT.PDS)GO TO 240

```

```

      PDS = PDS - XM(I)
230  STT = STT + ST(I)
      I = NST
240  NN = NRG(I)
      LLREG = ISV(NN)
      NST = 0
      IF(LLREG.LE.0) GO TO 300
      NST = 1
      SGR = TRB(I)
      IF(SGR.LE.0.0)GO TO 250
      ST(I) = PDS/TRAF(I)
      PDT = PDT/TRAF(I)
      GO TO 260
250  CONTINUE
      ARG = TRAF(I) - PDS*SGR
      PDT = PDT/ARG
      ST(I) = -ALDG(ARG/TRAF(I))/SGR
260  STT = STT + ST(I)
      PDT = PDT*PDT*STT**2
300  RETURN
      END
*IBFIC ARROWV M94/2,XR7
SUBROUTINE ARROW(X,S,C,Y)
DIMENSION X(3),C(3),Y(3)
DO 10 I=1,3
10  Y(I) = X(I) + S*C(I)
RETURN
END
*IBFIC QSTARV M94/2,XR7
CQSTAR *RANDOM SELECTION OF INITIAL DIRECTION VECTOR FOR FASTER*.JORDAN
FUNCTION QSTAR(INN,X,C)
COMMON H(I)
COMMON/LIMITS/NSSTORE,NERRDR,NSMAX,NMAX,NRMAX,NBMAX,NSTMAX,
1  NUNITX,NIMAX,NMMAX,NORDER,NODOWN,NELAS,NTRANS,
2  NNMXX,NCMAX,NFMAX,NVMOO,NCMAX,NLMXX,NTMAX,
3  NDMAX,NGHOD,NQENT,NDHOD,NPDMAX,NPDMOD,NSDMAX,
4  NVDMAX,NVDMOD,NPOINT,MODELP,MODELO,MODELU,MODELV,
5  NPRINT,NUNITS,NUMBER,WALIDE
COMMON/INDEXS/INTP ,IAZ ,IISV ,IMTL ,IRHO ,IXR ,IELL ,
1  IELM ,IAE ,IDR ,INSG ,INPC ,IJSN ,IJSX ,
2  IZIR ,ISUV ,IATN ,IATM ,IESB ,ISSH ,IDEN ,
3  INTG ,IELF ,IFGW ,ITOS ,IDOV ,ITDR ,IIDS ,
4  IVOL ,ICOT ,IXCT ,IRSI ,ALP ,IVMD ,IGIM ,
5  IAIM ,IALP ,IALH ,IA ,INS ,IVEE ,IVAL ,
6  ISM ,ISPE ,IATO ,IRSP ,ISGT ,IIOE ,IIOI ,
7  ISGS ,NEXT ,IXMP ,ISGR ,IMS ,IES ,IMC ,

```

FAST2906
FAST2907
FAST2908
FAST2909
FAST2910
FAST2911
FAST2912
FAST2913
FAST2914
FAST2915
FAST2916
FAST2917
FAST2918
FAST2919
FAST2920
FAST2921
FAST2922
FAST2923
FAST2924
FAST2925
FAST2926
FAST2927
FAST2928
FAST2929
FAST2930
FAST2931
FAST2932
FAST2933
FAST2934
FAST2935

FAST2937
FAST2938
FASA2939
FASA2939
FASC2939
FASC2939
FAST2940
FAST2941
FAST2942
FAST2943
FAST2944
FAST2945
FAST2946
FAST2947
FAST2948
FAST2949
FAST2950

FAST2951
FAST2952

FAST2953
FAST2954

FAST2955
FAST2956
FAST2957
FAST2958
FAST2959
FAST2960
FAST2961
FAST2962
FAST2963
FAST2964
FAST2965
FAST2966
FAST2967
FAST2968
FAST2969
FAST2970
FAST2971
FAST2972
FAST2973
FAST2974
FAST2975
FAST2976
FAST2977
FAST2978
FAST2979
FAST2980
FAST2981
FAST2982
FAST2983
FAST2984
FAST2985
FAST2986
FAST2987
FAST2988
FAST2989
FAST2990
FAST2991
FAST2992
FAST2993
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 = QSTOUM*NN,X,C,AXMAX,5,1,H(IISV),H(INPC),H(IVEE),H(IVMD), FAST2998
1     H(IALP),H(IXTR),H(INSGI) FAST2999
      RETURN FAST3000
      END FAST3001
$1BFTC OSTDM1 M94/2,XR7 FAST3002
COSTUM*RANDOM SELECTION OF INITIAL DIRECTION VECTOR FOR FASTER*JORDANFAST3003
FUNCTION QSTOUM(INN,X,C,L1,L7,L3,ISV,NPC,VEE,VMD,ALP,IXTR,NSG) FAST3004
  DIMENSION X(3),C(3) FAST3005
  DIMENSION ISV(1),NPC(1,5,1),VEE(1,1,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 ,BOC(3),ATA ,ATB ,ATC , FAST3009
1  ATD ,ATO ,AT ,BT ,AS ,BS ,JMN ,JMAX , FAST3010
2  KNTH ,KNAX ,JBAR ,NZERO FAST3011
  COMMON/LLSREG/LLREG FAST3012
  N = ISVNN) FAST3013
  N = LLREG FAST3014
  PDA = 1.0 FAST3015
  DO 10 I=4,5 FAST3016
  J = NPC(I,N) FAST3017
10 PDA = PDA*SAMPLE(VEE(1,1,N),VEE(J,I,N),VMD(I,N),ALP(I,N),VB(I)) FAST3018
  D(3) = VB(5) FAST3019
  SPH = SQRT(1.0 - C(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 = MSG(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 QSTOUM = PDA FAST3048
      RETURN FAST3049
      END FAST3050
$1BFTC USTARX M94/2,XR7 FAST3051
CUSTAR *RANDOM SELECTION OF SCATTERING POINT FOR PROGRAM FASTER*JORDANFAST3052
FUNCTION USTARITTT,XXX,YYY,ZZZ,MNM,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,NXMAX ,NMAX ,NMAX ,NMAX ,NMAX ,NSTMAX, FAST3057
1  NEMAX ,NNMAX ,NXMAX ,NXMAX ,NEMOD ,NXSECT,NUNITO, FAST3058
2  NUNITX,MINAX ,NNMAX ,NORDER,NDDWN ,INELAS,NTRANS, FAST3059
3  NNMAX ,NGMAX ,NPMAX ,NVMOD ,NGMAX ,NLMAX ,NPMAX, FAST3060
4  NDMAX ,NGMOD ,NPMENT,NMOD ,NPMAX ,NPMOD ,NSDMAX, FAST3061
5  NVDMAX ,NVDMOD ,NPCINT ,MODELQ ,MODELQ ,MODELQ ,FAST3062
6  NPRINT ,NUNITS ,NUMBER ,KALIDE FAST3063
  COMMON/INDEXS/INTP ,IAZ ,IISV ,IMTL ,IRMD ,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 ,IDV ,IIDR ,IDIS , FAST3067
4  IVOL ,ICDT ,ICDT ,IRSI ,IALP ,IVMD ,IGIM , FAST3068
5  IAIM ,IALM ,IALM ,IA ,INS ,IVEE ,IVAL , FAST3069
6  ISPW ,ISPE ,IATO ,IRSP ,ISGT ,IIDR ,IIDR , FAST3070
7  ISGS ,NEXT ,IXMP ,ISGR ,INS ,IES ,INC , 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 = MNM FAST3075
  STZ = VECTOR(YYY ,XXX,CP) FAST3076
  STS = STZ**2 FAST3077
  PSI = -COS(INE(CP,ZZZ))*STZ FAST3078
  CALL USTOUM(NEMOD,1,H(IRMD),H(IISV),H(IMTL),H(ISTG),H(IND), FAST3079
1  H(IV),H(IEI),ST,NRG) FAST3080
  NST = STC FAST3081
  MNM = NST FAST3082
  USTAR = PDT FAST3083
  RETURN FAST3084
  END FAST3085
$1BFTC USTOM1 M94/2,XR7 FAST3086
CUSTOM*RANDOM SELECTION OF SCATTERING POINT FOR PROGRAM FASTER*JORDANFAST3087
SUBROUTINE USTOUM(L1,L2,RHD,SSH,NTL,SGT,XN,TRA,TBB,ST,NRG) FAST3088

```

```

DIMENSION ST(1),NRG(1),RHO(1),SSH(1),MTL(1),SGT(1,1,2),TRA(1),
1 TRB(1),XMT(1)
COMMON/DUMUST/STZ ,STS ,PSE ,STC ,NST ,PDT
COMMON/OTHERS/RADIUS,XCT(3),DELTA ,BDC(3),ATA ,ATB ,ATC ,
1 ATD ,AT ,BT ,AS ,BS ,JMIN ,JMAX ,
2 XMIN ,XMAX ,JBAR ,NZERO
COMMON/LIMITS/NSTORE,NERROR,NSPAX ,NAPAX ,NRMAX ,HBMAX ,NSTMAX ,
1 NEMAX ,NVMAX ,NXMAX ,NXEMAX,NEMOD ,NXSECT,NUNITD ,
2 NUNITX,NIMAX ,NMPAX ,NORDER,NODOWN ,INELAS,NTRANS ,
3 NNMIX ,NGMAX ,NFMAX ,NVMOD ,NCMAX ,NLMAX ,NIMAX ,
4 NDMAX ,NMOD ,NOPEN ,NMODD ,NPDMAX ,NPDMOD ,NSEMEX ,
5 NVDMAX,NVDMOD,NPCINT,MODEL,P,MODELQ,MODELU,MODELV ,
6 NPRINT,NUNITS,NUMBER,KALIDE
IF(MODELU.EQ.0) GO TO 89
STDB = SQR(STS)
XMTT = 0.0
XNST = 0.0
TRAB = 1.0
STT = 0.0
FMP = 0.0
DOZ = 1.0/STS
IF(IPOINT.GT.0) GO TO 5
DOZ = 1.0
IF(STDB.LE.DELTA) GO TO 5
DOZ = 10.5*DELTA/STDB**2
DOZ = DOZ*(1.0 + DOZ)
5 DOM = DOZ
XMBB = 0.0
DO 20 N=1,NST
I = NRG(N)
J = MTL(1)
SGR = RHO(1)+SSH(JBAR)
IF(J.GT.0) SGR = SGR + SGT(JBAR,J)
STN = ST(N)
STT = STT + STN
STDS = STS + STT*(2.0*PST + STT)
STDA = STDB
STDB = SQR(STDS)
XNST = XNST + AT*SGR*STN
XMTT = XMTT + BT*SGR*STDB - STDA)
XMAA = XMBB
XMBB = XNST + XMTT
TRAA = TRAB
DOM = 1.0/STDS
IF(IPOINT.GT.0) GO TO 8
DOM = 1.0
IF(STDB.LE.DELTA) GO TO 8

```

FAST3089
FAST3090
FAST3091
FAST3092
FAST3093
FAST3094
FAST3095
FAST3096
FAST3097
FAST3098
FAST3099
FAST3100
FAST3101
FAST3102
FAST3103
FAST3104
FAST3105
FAST3106
FAST3107
FAST3108
FAST3109
FAST3110
FAST3111
FAST3112
FAST3113
FAST3114
FAST3115
FAST3116
FAST3117
FAST3118
FAST3119
FAST3120
FAST3121
FAST3122
FAST3123
FAST3124
FAST3125
FAST3126
FAST3127
FAST3128
FAST3129
FAST3130
FAST3131
FAST3132
FAST3133
FAST3134
FAST3135

```

DOM = 10.5*DELTA/STDB**2
DOM = DOM*(1.0 + DOM)
8 TRAB = DOM*EXP(-XNST - XMTT)/DOZ
XMN = 0.0
IF((SGR+STN).EQ.0.0) GO TO 10
TRA(N) = SGR*TRAA
IF(TRA(N).EQ.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 XN(N) = XMN
20 FMP = FMP + XMN
IF(FMP.EQ.0.0) GO TO 95
30 FMB = FMP*ARND(NRB)
STT = 0.0
DO 40 N=1,NST
IF(XN(N).GE.FMB) GO TO 50
STT = STT + ST(N)
40 FMB = FMB - XN(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
NST = 1
TRA(1) = STZ
STT = 0.0
FMT = 0.0
90 FMP = 0.0
DO 120 N=1,NST
I = NRG(N)
J = MTL(1)
SGR = RHO(1)+SSH(JBAR)

```

FAST3136
FAST3137
FAST3138
FAST3139
FAST3140
FAST3141
FAST3142
FAST3143
FAST3144
FAST3145
FAST3146
FAST3147
FAST3148
FAST3149
FAST3150
FAST3151
FAST3152
FAST3153
FAST3154
FAST3155
FAST3156
FAST3157
FAST3158
FAST3159
FAST3160
FAST3161
FAST3162
FAST3163
FAST3164
FAST3165
FAST3166
FAST3167
FAST3168
FAST3169
FAST3170
FAST3171
FAST3172
FAST3173
FAST3174
FAST3175
FAST3176
FAST3177
FAST3178
FAST3179
FAST3180
FAST3181
FAST3182

```

1F(J,GT,0)SGR = SGR + SGT(JBAR,J)
1F(INGT,GT,0)GO TO 100
1F(SGR,Eq,0.0)GO TO 110
XM(N) = ST(N)*SGRWANG
GO TO 120
100 ST = STT + ST(N)
  TRAI(N) = SORT(STS + STT*(2.0*P$1 + STT*P$1$1)
  TRB(N) = 0.0
  1F(SGR,Eq,0.0)GO TO 110
  TRB(N) = BT*SGR*(TRAI(N) - TRAI(N))
  FBT = FBT + TRB(N)
  XM(N) = TRB(N) + AT*SGR*ST(N)
  GO TO 120
110 XM(N) = 0.0
120 FMP = FMP + XM(N)
  1F(FMP,GT,0.0)GO TO 125
  NST = 0
  GO TO 195
125 AAA = EXP(-FMP)
  ARG = 1.0 - AAA
  VARG = 1.0 - ARG*RHNO(NHMB)
  FMA = -ALOG(VARG)
  FMB = FMA
  FMC = 0.0
  STC = 0.0
  DO 140 I=1,NST
    1F(FNB,LE,XM(I))GO TO 150
    FMB = FMB - XM(I)
    STC = STC + ST(I)
  140 FMC = FMC + TRB(I)
150 NN = NRG(I)
  NN = MTL(NN)
  NST=I
  SGP = RH(NN)*SSH(JBAR)
  1F(MN,GT,0)SGP = SGP + SGT(JBAR,MN)
  1F(INGT,GT,0)GO TO 160
  SGP = ANG*SGP
  DOK = 1F(MP*AAA/ARG + FMA - 1.0)/ANG
  STP = FMB/SGP
  STC = STC + STP
  GO TO 170
160 SGA = AT*SGP
  SGB = BT*SGP
  Q = (FMB + SGA*STC + SGB*TRAI(1))/SGB
  AQ = SGA/SGB
  FST = AQ*P2 - PS15
  BQ = (AQ*QCQ + PS11)/FST

```

```

HH = (BCQ*2 - (CQ*2 - STS)/VST)
1F(HH,LT,0,0) HH=0.0
HH=SQRT(HH)
STQ = BQ - HH
1F(STQ,LT,STC)STQ = BQ + HH
STC = STQ - STQ
STC = STQ
ARE = CQ - AQ*STC
FMC = FMC + SGB*ARE - TRA(1)
PSI = (PSI + PSI*STC)/ARE
ANG = AT + BT*PSI
SGP = ANG*SGP
1F(1) = STP
PDT = ARG*STC*27/(SGP*VARG*12.566371)
195 RETURN
END
$1BFTC VSTAR1 M94/2,RT
CVSTAR =RANDOM SELECTION OF DIRECTION VECTOR FOR PROGRAM FASTER*Y.JORDAN
FUNCTION VSTAR(CP,CC,C,AS,BS)
COMMON/LIMITS/NSTORE,NERROR,NSPAR,NMAX,NRMAX,NBMAX,NSTMAX,
1 NEMAX,NVMAX,NLMAX,NPEMAX,NEND,NXSECT,NUNIT,
2 NUNIT,NMAX,NMAX,NCRDER,NORDER,NINELAS,NTRANS
3 NMAX,NCMAX,NPEAX,NVNO,NCMO,NLMAX,NLMAX,NMAX
4 NDMAX,NACMD,NPCENT,NKMD,NPMOAX,NPMOD,NSDMAX
5 NVMAX,NWDPCD,NPCMT,MODEP,NMODEL,MODELU,MODELV,
6 NPRINT,NUNITS,NUMBER,KALIDE
7 IF(MODELV - 11 5,10,15
8 5 CONTINUE
9 VSTAR = VSTOUN(CP,C,C,BS,AS)
10 GO TO 20
11 VSTAR = VSTOUN(CP,CC,C,AS,BS)
12 GO TO 20
13 VSTAR = VSTOUN(CP,CC,C,AS + BS,0.0)
20 RETURN
END
$1BFTC VSTDM1 M94/2,RT
FUNCTION VSTDM(CP,CC,C,AS,BS)
DATA P1/3,16169/
1 DIMENSION CP(3),CC(3),C(3),ROT(3,3),D(3)
2 EQUIVALENCE (D(1),D1),(D(2),D2),(D(3),D3)
3 CALL ROTATG(CP,D1)
4 DO 10 I=1,3
5 D(I) = 0.0
6 DO 10 J=1,3
7 D(1,I) = D(1) + ROT(1,J)*CC(J)
8 DO 10 J=1,3
9 1F((ABS(D(1) + ABS(D(2)) - GT,0.0) THZ = ATAN2(D2,D1)

```

```

      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,-.85*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) + D1(J)*ROT(J,I)
      VSTDUM = PDA
      RETURN
      END
      $IBFTC SAMPLR M94/2,AR7
      C SAMPLE RANDOM SELECTION OF VARIABLE
      FUNCTION SAMPLE(VNM,VMX,VMD,ALP,VEE)
      IF(VMX.GT.VMNI)GO TO 10
      C DELTA FUNCTION (FOR 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 - VMNI) - 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*ALG(R)/ALP
      PDF = PGT/(R*ALP)
      30 SAMPLE = PDF
      RETURN
      END
      $IBFTC RANDOM M94/2,XR7
      FUNCTION RANNO(NMB)
      COMMON/CDC1B/IEMCDC
      DATA N,NMB/0,11111/
      IF(IEMCDC.GT.0) GO TO 10
      R = RANF(NNN)
      GO TO 20
  
```

FAST3278
 FAST3279
 FAST3280
 FAST3281
 FAST3282
 FAST3283
 FAST3284
 FAST3285
 FAST3286
 FAST3287
 FAST3288
 FAST3289
 FAST3290
 FAST3291
 FAST3292
 FAST3293
 FAST3294
 FAST3295
 FAST3296
 FAST3297
 FAST3298
 FAST3299
 FAST3300
 FAST3301
 FAST3302
 FAST3303
 FAST3304
 FAST3305
 FAST3306
 FAST3307
 FAST3308
 FAST3309
 FAST3310
 FAST3311
 FAST3312
 FAST3313
 FAST3314
 FAST3315
 FAST3316
 FAST3317
 FAST3318
 FAST3319
 FAST3320
 FAST3321
 FAST3322
 FAST3323

```

      10 CONTINUE
      IF(N.EQ.0) R = RANF(NNA)
      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 L14
      L16 OCT 002312421637
      L17 OCT 000000001737
      L20 OCT 20CG00000000
      END
      $DATA
  
```

FAST3325
 FAST3326
 FAST3327
 FAST3328
 FAST3329
 FAST3330
 FAST3331
 FAST3332