

325
5/14/64

NAA-SR-9233

COPY

MASTER

QUICKIE -
A COMPUTER PROGRAM FOR SPATIALLY INDEPENDENT
MULTIGROUP SLOWING-DOWN AND
THERMALIZATION CALCULATIONS

AEC Research and Development Report



ATOMICS INTERNATIONAL

A DIVISION OF NORTH AMERICAN AVIATION, INC.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

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

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Price \$1.25

Available from the Office of Technical Services
Department of Commerce
Washington 25, D. C.

NAA-SR-9233
MATHEMATICS AND
COMPUTERS
51 PAGES

QUICKIE -
A COMPUTER PROGRAM FOR SPATIALLY INDEPENDENT
MULTIGROUP SLOWING-DOWN AND
THERMALIZATION CALCULATIONS

By
M. BOLING
W. RHOADES

ATOMICS INTERNATIONAL

A DIVISION OF NORTH AMERICAN AVIATION, INC.
P.O. BOX 309 CANOGA PARK, CALIFORNIA

CONTRACT: AT(11-1)-GEN-8
ISSUED: APR 15 1964

DISTRIBUTION

This report has been distributed according to the category "Mathematics and Computers" as given in "Standard Distribution lists for Unclassified Scientific and Technical Reports" TID-4500 (26th Ed.) January 1, 1964. A total of 650 copies was printed.

ACKNOWLEDGMENT

The authors wish to acknowledge the assistance of W. J. Roberts, who suggested the method of solution for thermal fluxes; of G. A. Long, who assisted in programming and reporting an earlier version of QUICKIE; and of M. Hoffman, who developed the FLOPUN subroutine.

CONTENTS

	Page
Abstract	4
I. Introduction	5
II. Method	6
A. Basic Integral Flux Equations	6
B. Single Fissioning Isotope	7
C. Spectral Data	8
D. Few-Group Neutron Economy	8
E. Few-Group Cross Sections	9
III. Method of Solution	10
A. Macroscopic Cross-Section Compilation	10
B. Flux Solution	10
C. Special Calculations	12
IV. Data Input Description	14
A. Control Card	14
B. Material Cards	15
C. Buckling Cards	15
D. Flux Cards	16
E. Group Reduction Card	16
F. Library Tape	16
References	17
Appendix	17

ABSTRACT

In the case of an infinite homogeneous medium or a finite medium where buckling is known through calculation or physical considerations, the multigroup diffusion equations can be converted to spatially independent equations involving only integrated fluxes. These equations are quite simple to solve, giving the neutron energy spectrum, multiplication factor, and total reaction rates.

QUICKIE performs these calculations as well as calculating few-group cross sections and punching libraries for the spatially dependent diffusion and transport codes. Its advantages are simplicity of input data requirements and economy of operation.

I. INTRODUCTION

QUICKIE is a computer program designed to solve the multigroup neutron slowing down and thermalization equations without consideration of spatial dimensions. This calculation does not provide information about spatial distribution of neutron density, but it does provide an accurate neutron energy spectrum in the case of infinite homogeneous media, and also for a limited class of finite homogeneous media.

The effect of leakage, in the case of a finite medium, is included by means of an effective buckling. In the case of a bare core with flux vanishing at a single boundary, the geometric buckling is supplied as input data. If the flux is considered to vanish at the group dependent extrapolated boundaries, the core dimensions are supplied, and the bucklings are calculated internally. Homogeneous reactors reflected in one dimension are also satisfactorily treated using a reflector savings or effective buckling known from previous calculations.

The uses of QUICKIE include the study of thermal spectrum temperature coefficients, the effect of thermal poisons and fuel changes, and certain approximate criticality surveys involving size and composition. The program provides a neutron energy spectrum, fission, capture, and leakage spectra, few-group cross sections, and a few-group neutron economy table. In addition, energy and lethargy plots of the flux spectrum are provided. The cross-section data are read from tapes used with spatially dependent codes. The version of QUICKIE listed in the appendix uses the ULCER library tape,¹ and a version to use the DTF (FORTRAN DTK) tape is planned.

The advantages of QUICKIE are simplicity and economy. It may require as few as two cards of input data for a problem, as compared with 20 to 60 for a one-dimensional code. It requires no more than 6 sec of computer time for a calculation, as compared to 6 min for the corresponding one-dimensional code. For problems which can be treated satisfactorily by QUICKIE, these make it an attractive alternative to the spatial treatment.

II. METHOD

A. BASIC INTEGRAL FLUX EQUATIONS

The multigroup diffusion equations for a homogeneous reactor operating at steady state can be written:²

$$\Sigma_a^i \phi^i(\bar{r}) - D^i \nabla^2 \phi^i(\bar{r}) + \sum_j \phi^j(\bar{r}) \Sigma_{ji} = \frac{1}{\lambda} \sum_N \chi_N^i \sum_j (\nu \Sigma_f)_N^j \phi^j(\bar{r}) \quad \dots (1a)$$

$$+ \sum_j \phi^j(\bar{r}) \Sigma_{ij} \quad ,$$

$$\lambda = \int \sum_N \sum_j (\nu \Sigma_f)_N^j \phi^j(\bar{r}) d\bar{r} \quad , \quad \dots (1b)$$

where

$\bar{r}$ = spatial variable

Σ_a^i = absorption cross section, ith energy group

D^i = diffusion coefficient, ith energy group

ϕ^i = neutron flux, ith energy group

$(\nu \Sigma_f)_N^i$ = fission cross section, ith energy group, N_{th} element

χ_N^i = fission source spectrum, ith energy group, N_{th} element, normalized to 1 neutron

Σ_{ji} = scattering cross section, group i to group j

λ = multiplication factor.

If we integrate this over the reactor volume, we obtain the reaction rate for the system:

$$\Sigma_a^i \int \phi^i(\bar{r}) d\bar{r} - D^i \int \nabla^2 \phi^i(\bar{r}) d\bar{r} + \sum_j \Sigma_{ji} \int \phi^j(\bar{r}) d\bar{r}$$

$$= \frac{1}{\lambda} \sum_N \chi_N^i \sum_j (\nu \Sigma_f)_N^j \int \phi^j(\bar{r}) d\bar{r} + \sum_j \Sigma_{ij} \int \phi^j(\bar{r}) d\bar{r} \quad . \quad \dots (2)$$

Defining:

$$\Phi^i \equiv \int \varphi^i(\bar{r}) d\bar{r} \quad (\text{integral flux}) \quad , \quad \dots (3)$$

and:

$$(B^2)^i \equiv \frac{\int \nabla^2 \varphi^i(\bar{r}) d\bar{r}}{\int \varphi^i(\bar{r}) d\bar{r}} \quad (\text{effective buckling}) \quad , \quad \dots (4)$$

we have:

$$\Sigma_a^i \Phi^i + D^i (B^2)^i \Phi^i + \sum_j \Sigma_{ji} \Phi^j = \frac{1}{\lambda} \sum_N \chi_N^i \sum_j (\nu \Sigma_f)_N^j \Phi^j + \sum_j \Sigma_{ij} \Phi^j \quad . \quad \dots (5)$$

Now defining:

$$\Sigma_T^i \equiv \Sigma_a^i + D^i (B^2)^i + \sum_{j \neq i} \Sigma_{ji} \quad (\text{total cross section}) \quad , \quad \dots (6)$$

$$S^i \equiv \frac{1}{\lambda} \sum_N \chi_N^i \sum_j (\nu \Sigma_f)_N^j \Phi^j \quad (\text{group source}) \quad , \quad \dots (7)$$

the equations are:

$$\Sigma_T^i \Phi^i = S^i + \sum_{j \neq i} \Sigma_{ij} \Phi^j \quad \begin{array}{l} i = 1, 2, \dots, N_g \\ j = 1, 2, \dots, N_g \end{array} \quad . \quad \dots (8)$$

These are a set of N_g simultaneous equations which completely determine the integral fluxes, providing $(B^2)^i$ is known. Since λ involves the flux spectrum, and is used in calculating the source, the set cannot easily be solved explicitly. An iterative process, starting with a guess of S^i , calculating Φ^i and λ , and then repeating the calculation of S^i is used. This converges in about three iterations, and is not objectionable.

B. SINGLE FISSIONING ISOTOPE

In the case where a single typical fission source is adequate, the fission source supplied as a starting guess (at the beginning of the ULCER tape) is used, and the flux solution is made using this starting guess for S^i , and calculating Φ^i directly. No iterations are required in this case.

C. SPECTRAL DATA

The differential flux, fission, absorption, and leakage spectra are calculated directly from the flux and the energy interval, $(\Delta E)^i$, for the i th energy group:

$$p^i \equiv \Phi^i / (\Delta E)^i \quad (\text{flux spectrum}) \quad , \quad \dots (9a)$$

$$f^i = p^i (\nu \Sigma_f)^i \quad (\text{fission spectrum}) \quad , \quad \dots (9b)$$

$$a^i = p^i \Sigma_a^i \quad (\text{absorption spectrum}) \quad , \quad \dots (9c)$$

$$l^i = p^i D^i (B^2)^i \quad (\text{leakage spectrum}) \quad . \quad \dots (9d)$$

D. FEW-GROUP NEUTRON ECONOMY

The few-group neutron economy data are calculated using a set of one to six-group breakpoints, $M_1, M_2, \dots, M_n$ which refer to the last multigroup to be included in each few group. The breakpoint M_0 is understood to be 0, and M_n is normally the last multigroup. The few-group flux, fission, absorption and leakage are calculated:

$$P^n = \sum_{i=M_{n-1}+1}^{M_n} \Phi^i \quad (\text{few group flux}) \quad , \quad \dots (10a)$$

$$F^n = \sum_{i=M_{n-1}+1}^{M_n} \Phi^i (\nu \Sigma_f)^i \quad (\text{few group fission}) \quad , \quad \dots (10b)$$

$$A^n = \sum_{i=M_{n-1}+1}^{M_n} \Phi^i \Sigma_a^i \quad (\text{few group absorption}) \quad , \quad \dots (10c)$$

$$L^n = \sum_{i=M_{n-1}+1}^{M_n} \Phi^i D^i (B^2)^i \quad (\text{few group leakage}) \quad . \quad \dots (10d)$$

E. FEW-GROUP CROSS SECTIONS

The few-group cross sections are then calculated by direct flux averaging:

$$(\overline{\nu\Sigma_f})^n = F^n/P^n \quad , \quad \dots(11a)$$

$$(\overline{\Sigma_a})^n = A^n/P^n \quad , \quad \dots(11b)$$

$$(\overline{DB^2})^n = L^n/P^n \quad , \quad \dots(11c)$$

$$(\overline{D})^n = L^n / \sum_{i=M_{n-1}+1}^{M_n} \Phi^i \left| (B^2) \right|^i \quad , \quad \dots(11d)$$

$$(B^2)^n = (DB^2)^n / (\overline{D})^n \quad , \quad \dots(11e)$$

$$\Sigma_{nn'} = \sum_{i=M_{n'-1}+1}^{M_{n'}} \Phi^i \sum_{j=M_{n-1}+1}^{M_n} \Sigma_{ji} / P^n \quad . \quad \dots(11f)$$

III. METHOD OF SOLUTION

A. MACROSCOPIC CROSS-SECTION COMPILATION

The solution of the first problem starts by reading from the library tape the group structure information and an initial guess for the source spectrum. The first element on the library tape is read, and the list of elements required to form the cross sections is scanned to see if the element is needed. If so, the microscopic cross sections are multiplied by the number density to form macroscopic cross sections. If not, the next element is read and treated similarly. Each element on the tape is examined, its contribution to the macroscopic cross section, if any, being added to those previously found, until all the required elements have been located. Then the tape is immediately re-wound for later use. This system is about as efficient as any, and in addition is quite simple.

The total cross section Σ_T^i and diffusion coefficient are constructed from the macroscopic data, and the scatter cross sections, Σ_{ji} are assembled.

The source, S^i , is set equal to the initial source guess.

B. FLUX SOLUTION

The solution for the fluxes (Equation 8) can now be represented in matrix form:

$$\begin{bmatrix} -\Sigma_T^1 & \Sigma_{1,2} & \cdots & \Sigma_{1,N_g} \\ \Sigma_{2,1} & -\Sigma_T^2 & & \\ \vdots & & \ddots & \\ \Sigma_{N_g,1} & & & -\Sigma_{N_g,N_g} \end{bmatrix} \begin{bmatrix} \Phi^1 \\ \vdots \\ \Phi^{N_g} \end{bmatrix} = - \begin{bmatrix} S^1 \\ \vdots \\ S^{N_g} \end{bmatrix} \quad \dots (12)$$

In practice, upscatter to groups of index less than a limit N_{th} is prohibited by arranging the library such that:

$$\Sigma_{ji} = 0 \text{ for } \begin{cases} j < N_{th} \\ \text{and } j < i \end{cases}$$

The solution for T is thus reduced to:

$$\begin{bmatrix} -\Sigma_T^1 & 0 & \cdots & 0 \\ \Sigma_{2,1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{N_{th},1} & \cdots & -\Sigma_T^{N_{th}} & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ \Sigma_{N_g,1} & \cdots & \Sigma_{N_g,N_{th}} & \cdots & -\Sigma_T^{N_g} \end{bmatrix} \begin{bmatrix} \Phi^1 \\ \vdots \\ \Phi^{N_{th}} \\ \vdots \\ \Phi^{N_g} \end{bmatrix} = - \begin{bmatrix} S^1 \\ \vdots \\ S^{N_{th}} \\ \vdots \\ S^{N_g} \end{bmatrix}, \quad \dots (13)$$

The solution for $\Phi^1 \cdots \Phi^{N_{th}-1}$ can now be made explicitly:

$$\Phi^1 = S^1 / \Sigma_T^1, \quad \dots (14a)$$

$$\Phi^i = \left[S^i + \sum_{j=1}^{i-1} \Phi^j \Sigma_{ij} \right] / \Sigma_T^i \quad 1 < i < N_{th}. \quad \dots (14b)$$

A thermal source is then constructed:

$$S_{th}^i = S^i + \sum_{j=1}^{N_{th}-1} \Phi^j \Sigma_{ij} \quad N_{th} \leq i \leq N_g, \quad \dots (15)$$

and the thermal fluxes are found by:

$$\begin{bmatrix}
 -\Sigma_T^{Nth} & \Sigma_{Nth, Nth+1} & \dots & \Sigma_{Nth, Ng} \\
 \vdots & \vdots & \ddots & \vdots \\
 \Sigma_{Nth+1, Nth} & -\Sigma_T^{Nth+1} & & \\
 \vdots & \vdots & \ddots & \vdots \\
 \Sigma_{Ng, Nth} & \dots & \dots & -\Sigma_T^{Ng}
 \end{bmatrix}
 \begin{bmatrix}
 \Phi^{Nth} \\
 \vdots \\
 \Phi^{Ng}
 \end{bmatrix}
 = -
 \begin{bmatrix}
 S_{Th}^{Nth} \\
 \vdots \\
 S_{Th}^{Ng}
 \end{bmatrix}
 \quad \dots (16)$$

This is solved by a standard library routine for Φ^i .

This completes the flux solution, and λ is then calculated by Equation 1b. If the "chi-option" is not requested, the program performs the final edit using Equations 9, 10, and 11, and calls in the next problem. If the "chi-option" is requested, the source S^i is recomputed by Equation 7, and the flux solution and λ calculation repeated until convergence is reached on the i th iteration, according to:

$$\left| \frac{\lambda_i - \lambda_{i-1}}{\lambda_i} \right| < 0.00001 \quad .$$

When convergence is reached, the final edit is taken as usual.

C. SPECIAL CALCULATIONS

After the edit is completed, the source spectrum is replaced with the leakage spectrum. If the "reflector option" is requested on the next problem, this leakage spectrum will be used in lieu of the usual fission source spectrum. Thus, by supplying the composition of a reflector material for the next problem, a reflector spectrum calculation, useful to obtain reflector few-group cross sections, can be performed.

An option is provided to read in part or all of the fluxes in lieu of the usual flux calculation. This is useful for certain shielding cross-section calculations. Another option allows punching of multigroup or few-group libraries for various computer codes. The use of these options is covered in the Data Input Description section of this report.

IV. DATA INPUT DESCRIPTION

The QUICKIE program uses a library which gives microscopic cross sections for various elements used. Supplying the proper QUICKIE data along with this library, it is possible to perform a multigroup flux calculation for a one-region system. QUICKIE can also do a few-group cross-section calculation, a reflector spectrum calculation, an elemental fission spectrum calculation (chi-option), and gives CRT output.

The CRT output includes a histogram of $\Phi/\Delta u$ vs lethargy and a histogram of $\Phi/\Delta E$ vs energy for the thermal groups.

The following input is required.

A. CONTROL CARD

Col: 1-60 Title — Alphanumeric information

Col: 61-72 IC(1) through IC(6) in 6I2 format. This is control data and is used as follows:

IC(1) = -1: flux read from cards. A set of group-wise fluxes can be read from cards to enable the user to use his own flux spectrum. A blank or 0 flux for a group will cause the flux calculated in the previous problem to be used for that group.

= 0: the code will calculate its own flux

= 1: the flux from the previous case will be used for all groups

= 2: does a reflector spectrum calculation

= 3: the chi-option will be used

IC(2) = -1: punches the macroscopic cross sections onto cards which can be used to make an ULCER multigroup library

= 0: no punch

= +1: calculates few-group data (≤ 6) - reads new few-group structure

= +2: calculates few-group data (≤ 6) - uses previous few-group structure

IC(3) = -N: read over only those materials which are changing on the first N material cards, ignoring blank element names.

= 0: use previous material

= N: read N material cards. No element names from the previous case are retained.

IC(4) = 0: code will set $B^2 = 1.0 \times 10^{-20}$

= +1: read in a single B^2

= +2: read in groupwise B^2

= +3: use B^2 from previous problem

= -1: code will calculate B^2 for a plane geometry

= -2: code will calculate B^2 for a cylinder geometry

= -3: code will calculate B^2 for a spherical geometry

IC(5) = 0: D will be calculated as $1/(3 \Sigma_{tr})$

= 1: D will be calculated as $1/3(1-0.4 \Sigma_a / \Sigma_{tr})^2 \Sigma_{tr}$

IC(6) = -1: plots Φ/Δ vs lethargy (all groups) and $\Phi/\Delta E$ vs energy (for the thermal groups only)

= 0: no plots

= +1: plots only $\Phi/\Delta E$ vs energy (thermal groups)

B. MATERIAL CARDS

There will be N material cards which include the element names and concentrations. The format for these cards is 4(A6, F12.0). If IC(3) is negative, blank element names on the card will not change the element names previously appearing in those positions, and information in positions not specified will be retained. If IC(3) is positive, blanks will remove the previous element names, and information in positions not specified will be removed. If IC(3) = 0, no material cards are read and the previous element list is used.

C. BUCKLING CARDS

If IC(4) = 0 or 3 no B^2 card is read.

= +1: read in one B^2 card with group-independent B^2
 = +2: read in one B^2 for each group in 4(6X, F12.0)
 format. Since there can be 4 per card, there will be $N_g/4$ cards
 read.

If IC(4) is negative, only one card is required. The first piece of data is the calculated buckling option. If this value is 0, the code will calculate the standard buckling.² If this value is 1, the code calculates a curvature-corrected buckling.² For a plane case, the X, Y, and Z dimensions are required. For a cylindrical case, the radius and height dimensions are required. For the spherical case, the radius is required. All of the data in this section is read on the 4(6X, F12.0) format.

D. FLUX CARDS

If the fluxes are to be read from cards they are entered here. A flux for each group is read and the format used is 4(6X, F12.0).

E. GROUP REDUCTION CARD

This card is required only if IC(2) = +1. Two to 6 few-groups may be specified. The last multigroup to be included in each few group is entered in columns 1-36 in 6I6 format, followed by the punch option in the I6 format.

This option enables the user to get few-group data punched on cards in the various library formats used by FAIM, ULCER, and FLOCO. In the case of ULCER, a value for N_{th} (number of first thermal group) is required. This is also in the I6 format.

Col. 1-6	7-12	13-18	19-24	25-30	31-36	37-42	43-48
← few group limits →						punch option	N_{th} for
						= 0 no punch	ULCER
						= 1 FAIM	punch
						= 2 ULCER	
						= 3 FLOCO	

F. LIBRARY TAPE

A cross section tape prepared by the ULCER library program¹ must be mounted as logical tape 9.

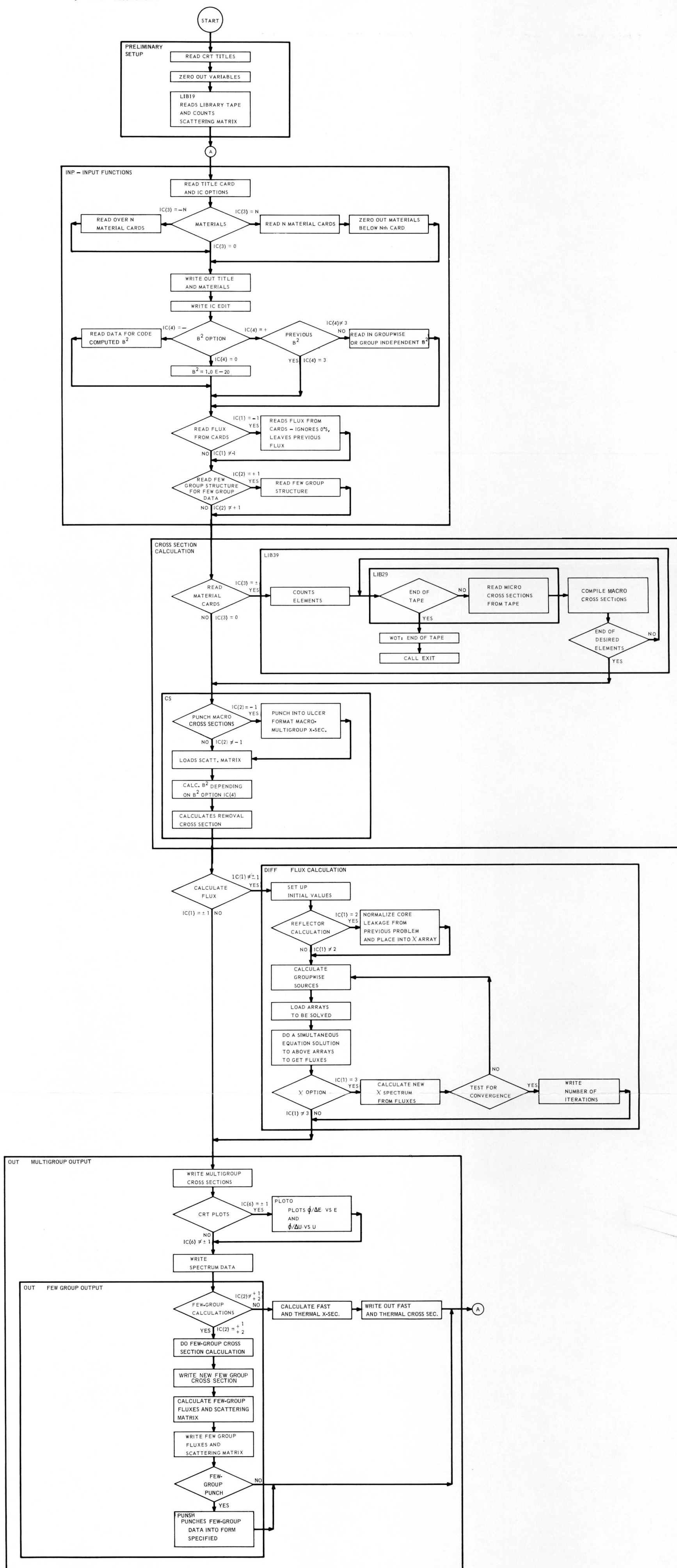
REFERENCES

1. D. Vargofcak, "The ULCER Code - A Multigroup, One-Dimensional, Diffusion Equation Code with Upscatter," NAA-SR-7146 (1962)
2. D. C. Baller, "The FAIM Code - A Multigroup One-Dimensional, Diffusion Equation Code," AMTD-118 (1962)

APPENDIX

This appendix contains:

- a. QUICKIE Block Diagram
- b. Listing of QUICKIE
- c. Sample Problem Input
- d. Sample Problem Output




```

* LABEL(QKEIII) 00000020
C QUICKIE III BY W.A.RHOADES AND M.A.BOLING 6-12-63 00000030
  DIMENSION 00000040
1 ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),00000050
2 FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40), 00000060
3 IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40), 00000070
4 STGT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6), 00000080
5 CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),00000090
6 C8(15),C9(15),IL01(5),SCHI(40) 00000100
7,AEXX(9),OEXX(9),ALXX(9),OLXX(9),ENGG(40),U(40),XXX(80),YYY(80) 00000105
  COMMON 00000110
1 IBLANK,N0G,NDSO,NDSU,NTH,LEM,AW,VUZERG,B,CHIIND,AFN,JZD,JZU,MX, 00000120
2 DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STGT,DNS,UPS,ENG,CHI,SSF,SSA,SST00000130
3,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,BOP,X7,Y7,Z7,R8,Z8,R9, 00000140
4 OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL0100000150
5,N5,N6,N9,I6M,SCHI,AEXX,OEXX,ALXX,OLXX,ENGG,U,XXX,YYY 00000160
  EQUIVALENCE 00000162
1 (IBLANK,ABLANK) 00000164
  00000170
  N5 = 5 00000180
  N6 = 6 00000190
  N9 = 9 00000200
  READ INPUT TAPE N5,20,AEXX,OEXX,ALXX,OLXX 00000202
20 FCRMAT(9A6) 00000204
  DO 10 I=1,40 00000210
10 B2(I) = 0.0 00000220
  ABLANK = 1H 00000230
  DO 12 I=1,20 00000240
  DI(I) = 0.0 00000250
12 NAME(I) = IBLANK 00000260
  00000270
C CALCULATE MACROSCOPIC CROSS SECTIONS 00000280
  00000290
11 CALL LIB19 00000300
  CALL INP 00000310
  IF(IC(3))18,14,18 00000320
18 CALL LIB39 00000330
14 CALL CS 00000340
  00000350
C CALCULATE FLUXES 00000360
  00000370
  IF(XABSF(IC(1)) -1 ) 16,15,16 00000380
16 CALL DIFF 00000390
  00000400
C CALCULATE OUTPUT DATA 00000410

```

15	CALL OUT	00000420
	GC TO 11	00000430
	END	00000440
*	LABEL(LIB19)	00000450
	SUBROUTINE LIB19	00000460
	DIMENSION	00000470
	1 ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),	00000480
	2 FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40),	00000490
	3 IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40),	00000500
	4 STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6),	00000510
	5 CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),	00000520
	6 C8(15),C9(15),IL01(5),SCHI(40)	00000530
	COMMON	00000540
	1 IBLANK,NCG,NDS,NDSU,NTH,LEM,AW,VUZERG,B,CHIIND,AFN,JZD,JZU,MX,	00000550
	2 DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST	00000560
	3,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,BOP,X7,Y7,Z7,R8,Z8,R9,	00000570
	4 OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL01	00000580
	5,N5,N6,N9,I6M,SCHI	00000590
		00000600
		00000610
C	READS LIBRARY TAPE AND COUNTS SCATTERING MATRIX	00000620
	REWINDN9	00000630
	READ TAPEN9,NCG,NDS,NDSU,NTH	00000640
	READ TAPEN9,(ENG(J),J=1,NCG)	00000650
	READ TAPE N9,(SCHI(J),J=1,NCG)	00000660
	JZD=0	00000670
	DC 20 J=2,NCG	00000680
	JM1=J-1	00000690
	DC 20 I=1,JM1	00000700
	IF(J-NDS-I)30,30,20	00000710
30	JZD = JZD + 1	00000720
20	CONTINUE	00000730
	JZU=0	00000740
	JM1=NCG-1	00000750
	DC 40 J=NTH,JM1	00000760
	JP1 = J + 1	00000770
	DC 40 I=JP1,NCG	00000780
	IF(I - NDSU - J)140,140,40	00000790
140	JZU = JZU + 1	00000800
40	CONTINUE	00000810
	RETURN	00000820
	END	00000830
*	LABEL(INP)	00000840
	SUBROUTINE INP	00000850
		00000860

	DIMENSION	00000870
1	ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),	00000880
2	FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40),	00000890
3	IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40),	00000900
4	STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6),	00000910
5	CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),	00000920
6	C8(15),C9(15),IL01(5),SCHI(40)	00000930
	COMMON	00000940
1	IBLANK,NCG,NDSD,NDSU,NTH,LEM,AW,VUZERG,B,CHIIND,AFN,JZD,JZU,MX,	00000950
2	DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST	00000960
3	,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,BOP,X7,Y7,Z7,R8,Z8,R9,	00000970
4	OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL01	00000980
5	,N5,N6,N9,I6M,SCHI	00000990
	DIMENSION NA(20),C(40)	00001000
		00001010
C	READS INPUT AND STORES BUCKLING IN PROPER LOCATIONS	00001020
		00001030
150	READ INPUT TAPENS , 7 , TITLE , IC	00001040
7	FORMAT (10A6,6I2)	00001050
	ICI =XABSF(IC(3)*4)	00001060
	IF(IC(3))70,50,60	00001070
70	READ INPUT TAPE N5,6,(NA(I),D(I),I=1,ICI)	00001080
	DO 80 I=1,ICI	00001090
	IF(NA(I) - IBLANK)90,80,90	00001100
90	NAME (I) = NA(I)	00001110
	DI(I) = D(I)	00001120
80	CONTINUE	00001130
	GO TO 50	00001140
60	DO 95 I=1,20	00001150
	NAME(I) = IBLANK	00001160
95	DI(I) = 0.0	00001170
	READ INPUT TAPE N5,6,(NAME(I),DI(I),I=1,ICI)	00001180
6	FORMAT (4(A6,F12.0))	00001190
50	WRITE OUTPUT TAPEN6 , 1 ,TITLE ,(I,NAME(I),DI(I),I=1,20)	00001200
1	FORMAT (1H1 10A6 /// (33X,I3,7X,A6,6X,1PE14.7 /))	00001210
	WRITE OUTPUT TAPEN6,551,(IC(I),I=1,6)	00001220
551	FORMAT(//4X,10HIC EDIT--6I6)	00001230
		00001240
1150	IC4 = IC (4)	00001250
	IF(IC4)101,120,102	00001260
101	N12 = - IC4	00001270
	GO TO (103, 104, 105), N12	00001280
103	READ INPUT TAPE N5,8,BOP,X7,Y7,Z7	00001290
	GO TO 20	00001300
104	READ INPUT TAPE N5,8,BOP,R8,Z8	00001310

GC TC 20	00001320
105 READ INPUT TAPE N5,8,B0P,R9	00001330
GC TC 20	00001340
	00001350
120 DC 122 I=1,N0G	00001360
OMEGA(I) = 0.0	00001370
122 B2(I) = 1.0E-20	00001380
GC TC 20	00001390
102 GC TC (30,10,20),IC4	00001400
30 READ INPUT TAPEN5 , 8 ,B	00001410
8 FORMAT (4(6X,F12.0))	00001420
DC 40 I = 1, N0G	00001430
40 B2 (I) = B	00001440
GC TC 20	00001450
10 READ INPUT TAPEN5 ,8 ,(B2 (I), I = 1, N0G)	00001460
	00001470
20 IF(IC(1)+1)320,310,320	00001480
310 READ INPUT TAPE N5, 8 ,(D(I),I=1,N0G)	00001490
DC 312 I=1,N0G	00001500
IF(D(I))314,312,314	00001510
314 PHI(I) = D(I)	00001520
312 CONTINUE	00001530
	00001540
320 IF(IC(2)-1)302,303,302	00001550
303 READ INPUT TAPE N5,301,(IL(I),I=1,6),I6M,(IL01(I),I=1,5)	00001560
301 FORMAT(12I6)	00001570
302 RETURN	00001580
END	00001590
* LABEL(LIB39)	00001600
SUBROUTINE LIB39	00001610
DIMENSION	00001620
1 ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),	00001630
2 FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40),	00001640
3 IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40),	00001650
4 ST0T(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6),	00001660
5 CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),	00001670
6 C8(15),C9(15),IL01(5),SCHI(40)	00001680
COMMON	00001690
1 IBLANK,N0G,NDSD,NDSDU,NTH,LEM,AW,VUZERG,B,CHIIND,AFN,JZD,JZU,MX,	00001700
2 CII,IC,B2,NAME,CI,TITLE,SF,SA,ST,ST0T,DNS,UPS,ENG,CHI,SSF,SSA,SST	00001710
3,CNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,B0P,X7,Y7,Z7,R8,Z8,R9,	00001720
4 OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL01	00001730
5,N5,N6,N9,I6M,SCHI	00001740
DIMENSION NAMEX(20)	00001750
	00001760

C COUNTS ELEMENTS AND COMPILES MACROSCOPIC CROSS SECTIONS

```

DC 30 I=1,40
SF ( I ) = 0.0
SA ( I ) = 0.0
ST ( I ) = 0.0
DC 30 J=1,40
SIG(J,I) = 0.0
AA( J,I) = 0.0
30 CHI(J,I) = 0.0
DC 34 I=1,780
32 DNS ( I ) = 0.0
34 UPS ( I ) = 0.0
NSUM = 0
DC 1 I = 1 , 20
NAMEX(I) = NAME(I)
IF(NAME(I) - IBLANK)3,1,3
3 NSUM = NSUM + 1
1 CONTINUE

16 CALL LIB29
IF( LEM - IBLANK )18,120,18
18 DC 5 I = 1 , 20
IF ( NAME (I) - LEM ) 5, 6, 5
6 IF ( NAMEX(I) -IBLANK)7, 5, 7
7 DII = DI ( I )
NAMEX(I) = IBLANK
NSUM = NSUM - 1

DC 10 J=1,N0G
SF(J) = SF(J) + SSF(J)*DII*VUZER0
SA(J) = SA(J) + SSA(J)*DII
10 ST(J) = ST(J) + SST(J)*DII
DC 15 J=1,JZD
15 DNS(J) = DNS(J) + DNSTR(J)*DII
DC 20 J=1,JZU
20 UPS(J) = UPS(J) + UPSTR(J)*DII
IF(CHIIND)22,24,22
22 DC 28 J=1, N0G
DC 28 K=1, N0G
28 CHI(K,J)= CHI(K,J)+VUZER0*SSF(K) * FCHI(J)* DII
GC TC 29
24 IF(VUZER0) 25,29,25
25 DC 26 J=1,N0G
DC 26 K=1,N0G

```

00001770
00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
00002000
00002010
00002020
00002030
00002040
00002050
00002060
00002070
00002080
00002090
00002100
00002110
00002120
00002130
00002140
00002150
00002160
00002170
00002180
00002190
00002200
00002210

```

26 CHI(K,J) = CHI(K,J) + VUZERO *SSF(K) * SCHI(J)*DII          00002220
29 IF(NSUM) 14,14,5                                             00002230
5  CONTINUE                                                     00002240
   GO TO 16                                                     00002250
14 REWINDN9                                                     00002260
114 RETURN                                                       00002270
120 WRITE OUTPUT TAPE N6,121,NAMEX                             00002280
121 FORMAT(//6X,55HTHE FOLLOWING ELEMENTS WERE NOT FOUND IN THE LIBRARY 00002290
1Y - // (6(6X,A6)) )                                           00002300
   CALL EXIT                                                    00002310
   STOP                                                         00002320
   END                                                         00002330
* LABEL(LIB29 )                                                 00002340
  SUBROUTINE LIB29                                             00002350
  DIMENSION                                                    00002360
1  ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780), 00002370
2  FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40), 00002380
3  IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40), 00002390
4  STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6), 00002400
5  CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6), 00002410
6  C8(15),C9(15),ILO1(5),SCHI(40)                             00002420
   COMMON                                                       00002430
1  IBLANK,NOG,NDSD,ND SU,NTH,LEM,AW,VUZERO,B,CHIIND,AFN,JZD,JZU,MX, 00002440
2  DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST 00002450
3  ,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,BOP,X7,Y7,Z7,R8,Z8,R9, 00002460
4  OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,ILO1 00002470
5  ,N5,N6,N9,I6M,SCHI                                         00002480
C   READS MICRO CROSS SECTIONS FROM LIBRARY TAPE              00002490
   EQUIVALENCE (THEEND,IHEEND)                                00002500
   THEEND = 6HTHEEND                                           00002510
   READ TAPEN9,LEM                                              00002520
   IF(LEM - IHEEND) 2,30,2                                     00002530
2  READ TAPEN9,AW,VUZERO,B,CHIIND                              00002540
   IF(VUZERO)5,10,5                                           00002550
5  READ TAPEN9,(SSF(J),J=1,NOG)                                00002560
10 READ TAPEN9,(SSA(J),J=1,NOG)                                00002570
   READ TAPEN9,(SST(J),J=1,NOG)                                00002580
   READ TAPEN9,(DNSTR(J),J=1,JZD)                              00002590
   READ TAPEN9,(UPSTR(J),J=1,JZU)                              00002600
   IF(CHIINC)15,20,15                                          00002610
15 READ TAPEN9,(FCHI(J),J=1,NOG)                               00002620
20 READ TAPEN9,AFN                                              00002630
   GO TO 25                                                     00002640

```

```

30 LEM = IBLANK                                00002670
25 RETURN                                        00002680
    END                                          00002690
    LABEL(CS      )                            00002700
    SUBROUTINE CS                               00002710
    DIMENSION                                   00002720
1  ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),00002730
2  FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40), 00002740
3  IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40), 00002750
4  STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6), 00002760
5  CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),00002770
6  C8(15),C9(15),IL01(5),SCHI(40)            00002780
    COMMON                                      00002790
1  IBLANK,N0G,NDSD,NDSU,NTH,LEM,AW,VUZER0,B,CHIIND,AFN,JZD,JZU,MX, 00002800
2  DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST00002810
3  ,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,B0P,X7,Y7,Z7,R8,Z8,R9, 00002820
4  OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL0100002830
5  ,N5,N6,N9,I6M,SCHI                        00002840
C  LOADS SCATTERING MATRICIES SIG AND AA WITH MACR0 CR0SS SECTIONS 00002850
C  INCLUDES OPTIONAL PUNCH OUTPUT 0FMACR0 CR0SS SECTIONS FOR ULCER 00002860
C  AND OPTION FOR CODE-CALCULATED BUCKLING 00002870
                                     00002880
    IF ( IC (2) + 1 ) 111 ,112,111            00002890
112 PUNCH 7,( SA(I) , I = 1 , N0G )           00002900
    PUNCH 7,( ST(I) , I= 1 , N0G )            00002910
    7 FORMAT( 6E12.5)                         00002920
    PUNCH 7,( DNS(I) , I = 1, JZD )           00002930
    PUNCH 7,( UPS(I) , I = 1, JZU )           00002940
    DO 9 I =1,5                               00002950
    9 PUNCH 8                                  00002960
    8 FORMAT (72H+++++00002970
1+++++00002980
111 IF(IC(3)) 150,50,150                      00002990
150 DO 10 I=1,40                              00003000
    DO 10 J=1,40                              00003010
    10 SIG(I,J)=0.0                          00003020
                                     00003030
    KK = C                                    00003040
    DO 20 J = 2 , N0G                        00003050
    JM1 = J - 1                              00003060
    DO 20 I = 1 , JM1                        00003070
    IF ( J - NDSC - I ) 30 , 30 , 20         00003080
30 KK= KK+ 1                                00003090
    SIG(I,J) = DNS(KK)                      00003100
                                     00003110

```

20	CONTINUE	00003120
	KK = 0	00003130
	JM1 = N0G - 1	00003140
	DC 40 J = NTH , JM1	00003150
	JP1 = J + 1	00003160
	DC 40 I = JP1 , N0G	00003170
	IF (I - NDSU - J) 140,140, 40	00003180
140	KK = KK + 1	00003190
	SIG(I,J) = UPS(KK)	00003200
40	CONTINUE	00003210
	IF(IC(5) - 1)50,62,50	00003220
62	DC 45 I = 1, N0G	00003230
45	ST(I) = ST(I) * (1.0 - 0.4*SA(I)/ST(I))**2	00003240
		00003250
		00003260
50	IF(IC(4))800,801,802	00003270
800	PIE = 3.1415927	00003280
	N0P1 = B0P	00003290
	DC 501 I = 1, N0G	00003300
501	OMEGA(I) = .710466 / ST(I)	00003310
	GC TO (121, 129, 124),N12	00003320
121	DC 122 I = 1,N0G	00003330
122	B2(I) = (PIE / (X7 + 2.0 * OMEGA(I)))*2+(PIE / (Y7 + 2.0 * OMEGA(I)))*2+(PIE / (Z7 + 2.0 * OMEGA(I)))*2	00003340
	GC TO 801	00003350
		00003360
124	DC 126 I = 1,N0G	00003370
127	IF (N0P1) 126, 126, 125	00003380
125	OMEGA(I) = OMEGA(I) / (1.0 + 0.710466/ST(I)/R9)	00003390
126	B2(I) = (PIE / (R9 + OMEGA(I)))*2	00003400
	GC TO 801	00003410
129	DC 132 I = 1,N0G	00003420
	B2(I) = (PIE / (Z8 + 2.0 * OMEGA(I)))*2	00003430
123	IF (N0P1) 132, 132 , 130	00003440
130	OMEGA(I) = OMEGA(I) / (1.0 + 0.355233/ST(I)/R8)	00003450
132	B2(I) = B2(I) + (2.405 / (R8 + OMEGA(I)))*2	00003460
	GC TO 801	00003470
802	DC 803 I = 1,N0G	00003480
803	OMEGA(I) = 0.0	00003490
801	CONTINUE	00003500
		00003510
C	CALCULATE REMOVAL C. S.	00003520
		00003530
	DC 60 I = 1 , N0G	00003540
	STOT(I) = SA(I) + B2(I)/(3.0*ST(I))	00003550
	DC 60 J=1,N0G	00003560

```

60 STOT(I) = STOT(I) + SIG(I,J)                                00003570
                                                                00003580
                                                                00003590
                                                                00003600
72 RETURN                                                       00003610
END                                                             00003620
* LABEL(DIFF )                                                 00003630
  SUBROUTINE DIFF                                             00003640
  DIMENSION                                                    00003650
1  ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),00003660
2  FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40), 00003670
3  IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40), 00003680
4  STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6), 00003690
5  CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),00003700
6  C8(15),C9(15),ILC1(5),SCHI(40)                             00003710
7  ,AEXX(9),CEXX(9),ALXX(9),CLXX(9),ENGG(40),U(40),XXX(80),YYY(80) 00003715
  COMMON                                                       00003720
1  IBLANK,NEG,NDSD,NDSDU,NTH,LEM,AW,VUZER0,B,CHIIND,AFN,JZD,JZU,MX, 00003730
2  CII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST00003740
3  ,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,B0P,X7,Y7,Z7,R8,Z8,R9, 00003750
4  OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL0100003760
5  ,N5,N6,N9,I6M,SCHI,AEXX,CEXX,ALXX,CLXX,ENGG,U,XXX,YYY       00003770
                                                                00003780
C   CALCULATES SOURCES AND FLUXES                             00003790
                                                                00003800
      ITER = 0                                                  00003810
      ITMAX = 100                                               00003820
      OLDK = 1.0E+32                                           00003830
7  IF(IC(1) -2) 1,2,1                                          00003860
2  SSUM = 0.0                                                  00003870
  DO 3 I = 1,NEG                                               00003880
3  SSUM = SSUM + SS(I)                                         00003890
  DO 4 I = 1,NEG                                               00003900
4  SCHI(I) = SS(I) /SSUM                                       00003910
1  DO 5 I=1,NEG                                                00003920
5  SS(I) = SCHI(I)                                             00003930
6  NM1 = NTH - 1                                              00003940
  DO 10 I = 1 , NM1                                           00003950
    PHI(I) = SS(I)/STOT(I)                                     00003960
    NP1 = I + 1                                                00003970
    DO 10 J = NP1 , NEG                                       00003980
10  SS(J) = SS(J) + PHI(I)*SIG(I,J)                           00003990
                                                                00004010
C   LOAD AA AND BB                                           00004020
98 DO 20 I=NTH,NEG                                           00004030

```

KI = I-NTH+1	00004040
DO 80 J= NTH,N0G	00004050
KJ= J -NTH + 1	00004060
80 AA(KJ,KI) =SIG(I,J)	00004070
BB(KI,1) = -SS(I)	00004080
20 AA(KI,KI) =-ST0T(I)	00004090
	00004100
F = 1.0	00004110
N = N0G - NTH + 1	00004120
DO 30 I = 1,N	00004130
30 IA(I) = 0	00004140
MX = XSIMEQF(40,N,1,AA,BB,F,IA)	00004150
31 DO 40 I = NTH,N0G	00004160
KI = I - NTH + 1	00004170
40 PHI(I) = AA(KI,1)	00004180
	00004190
IF(IC(1) - 3) 42,32,42	00004200
32 ITER = ITER + 1	00004210
XKEFF = C.0	00004220
DO 33 J=1,N0G	00004230
SCHI(J) =0.0	00004240
DO 34 K=1,N0G	00004250
34 SCHI(J) = SCHI(J) + CHI(K,J) * PHI(K)	00004260
33 XKEFF = XKEFF + SCHI(J)	00004270
DO 35 I=1,N0G	00004280
SCHI(I) = SCHI(I) /XKEFF	00004290
35 SS(I) = SCHI(I)	00004300
	00004310
EPS = 0LDK / XKEFF -1.0	00004320
IF(ABSF(EPS) - 0.00001) 41,41,36	00004330
36 IF(ITER - ITMAX) 37,43,43	00004340
37 0LDK = XKEFF	00004350
GO TO 6	00004430
	00004440
43 WRITE OUTPUT TAPE N6,44,EPS	00004450
44 FORMAT(//4X,44HWARNING, CONVERGENCE WAS NOT COMPLETE. ESP= E18.6)	00004460
41 WRITE OUTPUT TAPE N6,45,ITER	00004470
45 FORMAT(//I12,3X,25HITERATIONS WERE PERFORMED)	00004480
	00004490
42 GO TO (71,72,73),MX	00004500
71 WRITE OUTPUT TAPE N6,81,MX	00004510
81 FORMAT(//4X, 3HMX= I3 ,3X,26HS0LUTION WAS SUCCESSFUL)	00004520
74 RETURN	00004530
72 WRITE OUTPUT TAPE N6,82,MX	00004540
82 FORMAT(//4X, 3HMX= I3,3X,27HS0LUTION INDICATES OVERFLOW)	00004550

```

      GO TO 77                                00004560
73 WRITE OUTPUT TAPE N6,83,MX                00004570
83 FORMAT(/4X, 3HMX= I3,3X,37HSOLUTION INDICATES A SINGULAR MATRIX )00004580
77 CALL EXIT                                00004590
      STOP                                  00004600
      END                                  00004610
*      LABEL(OUT )                        00004615
      SUBROUTINE OUT                        00004620
      DIMENSION                            00004630
1  ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),00004640
2  FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40), 00004650
3  IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40), 00004660
4  STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6), 00004670
5  CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),00004680
6  C8(15),C9(15),IL01(5),SCHI(40)          00004690
7  ,AEXX(9),OEXX(9),ALXX(9),OLXX(9),ENGG(40),U(40),XXX(80),YYY(80) 00004695
      COMMON                                00004700
1  IBLANK,NCG,NDSO,NDSU,NTH,LEM,AW,VUZERO,B,CHIIND,AFN,JZD,JZU,MX, 00004710
2  CII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST00004720
3  ,DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,BOP,X7,Y7,Z7,R8,Z8,R9, 00004730
4  OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL0100004740
5  ,N5,N6,N9,I6M,SCHI,AEXX,OEXX,ALXX,OLXX,ENGG,U,XXX,YYY          00004750
                                00004760
C      PROBLEM RESULT EDIT                00004770
                                00004780
      DO 20 I = 1 , NCG                    00004790
      SIG1(I,1) = PHI(I)/ENG(I)             00004800
      SIG1(I,2) = 1.0/(3.0*ST(I))           00004810
20  SIG1(I,3) = B2(I)*SIG1(I,2)            00004820
      WRITE OUTPUT TAPEN6,3                00004830
3  FORMAT(1H0,1H1,7X,3HPHI,7X,13HPHI / DELTA E,8X,1HD,10X,10HNU SIGMA00004840
      X F,6X,7HSIGMA A,10X,3HDB2,8X,11HSIGMA TOTAL//) 00004850
      WRITE OUTPUT TAPEN6 , 2,(I,PHI(I),SIG1(I,1),SIG1(I,2),SF(I),SA(I),00004860
      XSIG1(I,3),STOT(I),I=1,NCG)          00004870
2  FORMAT(I3, 1P7E15.7 / )                00004880
                                00004881
      IF(IC(6)) 2007,2003,2007             00004882
2007 CALL PLCTO                             00004885
2003 TF=0.0                                00004890
      DO 24 I = 1 , NCG                    00004900
      SS(I) = PHI(I) * SIG1(I,3)           00004910
      SIG1(I,4) = SIG1(I,1) * SF (I)       00004920
      SIG1(I,5) = SIG1(I,1) * SA (I)       00004930
110 SIG1(I,6) = SIG1(I,1) * SIG1(I,3)     00004940
      24 TF = TF + PHI (I) * SF (I)        00004950

```

12	WRITE OUTPUT TAPEN6,4	00004960
4	FORMAT (1H1,5X,1H1,8X,5H0MEGA,12X,2H82,14X,3HCHI,11X,7HFISSION,7X,00004970	
	X10HABSORPTION,8X,7HLEAKAGE/1H ,61X,8HSPECTRUM,7X,8HSPECTRUM,9X,8HS00004980	
	XPECTRUM)	00004990
	WRITE OUTPUT TAPE N6,5,(I,0MEGA(I),B2(I),SCHI(I),SIG1(I,4),SIG1(I,00005000	
	X5),SIG1(I,6),I=1,N0G)	00005010
5	FORMAT (1H0,3X,I4,1P6E16.7)	00005020
	WRITE OUTPUT TAPEN6 , 201, TF	00005030
201	FORMAT (1H1,40X,7HK EFF =1PE15.7//)	00005040
		00005050
C	CALCULATE FEW GROUP CROSS SECTIONS	00005060
		00005070
1115	J = 1	00005080
	IX = 1	00005090
	IF(IC(2))118,118,116	00005100
116	DO 333 I=1,6	00005110
	IF(IL(I)) 333,334,333	00005120
333	CONTINUE	00005130
	I=7	00005140
334	I6 = I - 1	00005150
117	J1 = IL(IX)	00005160
	GO TO 115	00005170
118	I6=2	00005180
	I6M=0	00005190
	J1 = NTH - 1	00005200
115	TF = 0.0	00005210
	TA = 0.0	00005220
	TS = 0.0	00005230
	TDB2 = 0.0	00005240
	TB2 = 0.0	00005250
	TPHI = 0.0	00005260
	DO 10 I = J , J1	00005270
	TF = TF + PHI (I) * SF (I)	00005280
	TA = TA + PHI (I) * SA (I)	00005290
	TS = TS + PHI(I) * ST(I)	00005300
	TDB2 = TDB2 + PHI (I) * SIG1(I,3)	00005310
	TB2 = TB2 + PHI(I) * B2(I)	00005320
10	TPHI = TPHI + PHI (I)	00005330
	C1(IX) = TF	00005340
	C2(IX) = TA	00005350
	C3(IX) = TDB2	00005360
	CSA(IX) = TA/TPHI	00005370
	CVS6(IX) = TF/TPHI	00005380
	C4(IX) = TDB2/TB2	00005390
	C5(IX) = TS/TPHI	00005400

	C6(IX) = TCB2/TPHI	00005410
	C7(IX) = C6(IX)/C4(IX)	00005420
	CSTR(IX) = 1.0/(3.0*C4(IX))	00005430
126	IF(J1 - NCG)26,28,28	00005440
26	IF(IX - 6)27,28,28	00005450
27	IX = IX + 1	00005460
	J = J1 + 1	00005470
	IF(IC(2))127,127,117	00005480
127	J1 = NCG	00005490
	GO TO 115	00005500
		00005510
28	IF(IC(2))728,728,130	00005520
130	WRITE OUTPUT TAPEN6,302, (IL(I),I=1,I6)	00005530
302	FORMAT(1H0,6X,16HFEW GROUP LIMITS,5X,14HMULTIGROUPS 614)	00005540
728	WRITE OUTPUT TAPEN6,1212,(I,I=1,I6)	00005550
1212	FORMAT(////5X,12HGROUP- - - -(I23,5I15)///)	00005560
	WRITE OUTPUT TAPEN6,1213,(C1(IX),IX=1,I6)	00005570
1213	FORMAT(1H0,5X,7HFISSION 1P6E15.5)	00005580
	WRITE OUTPUT TAPEN6,1214,(C2(IX),IX=1,I6)	00005590
1214	FORMAT(1H0,2X,10HABSCRIPTION 1P6E15.5)	00005600
	WRITE OUTPUT TAPEN6,1215,(C3(IX),IX=1,I6)	00005610
1215	FORMAT(1H0,5X,7HLEAKAGE 1P6E15.5)	00005620
	WRITE OUTPUT TAPEN6,1216,(C4(IX),IX=1,I6)	00005630
1216	FORMAT (1H0,11X,1HD 1P6E15.5)	00005640
	WRITE OUTPUT TAPEN6,1217,(CVS6(IX),IX=1,I6)	00005650
1217	FORMAT(1H0,2X,10HNV SIGMA F 1P6E15.5)	00005660
	WRITE OUTPUT TAPEN6,1218,(CSA(IX),IX=1,I6)	00005670
1218	FORMAT(1H0,5X,7HSIGMA A 1P6E15.5)	00005680
	WRITE OUTPUT TAPEN6,1219,(C5(IX),IX=1,I6)	00005690
1219	FORMAT(1H0,4X, 8HSIGMA TR 1P6E15.5)	00005700
	WRITE OUTPUT TAPEN6,1220,(C6(IX),IX=1,I6)	00005710
1220	FORMAT(1H0,8X,4HD B2 1P6E15.5)	00005720
	WRITE OUTPUT TAPEN6,1221,(C7(IX),IX=1,I6)	00005730
1221	FORMAT(1H0,9X,3HB 2 1P6E15.5)	00005740
	WRITE OUTPUT TAPEN6,1222,(CSTR(IX),IX=1,I6)	00005750
1222	FORMAT(1H0,5X,7H1/(3*D) 1P6E15.5)	00005760
		00005770
C	CALCULATE FEW GROUP FLUXES AND SCATTERING MATRIX	00005780
		00005790
	IF(IC(2))128,128,740	00005800
740	IT = 1	00005810
	DO310 I = 1,I6	00005820
	IB = IL(I)	00005840
	PH(I) = 0.0	00005850
	DO 15 II = IT,IB	00005860

```

15 PH(I) = PH(I) + PHI(II)                                00005870
   JT = 1                                                  00005880
   DC320 J = 1,I6                                         00005890
   JB = IL(J)                                              00005910
   SIG1(I,J) = 0.0                                         00005920
   DC 30 II = IT,IB                                       00005930
   DC 30 JJ = JT,JB                                       00005940
30 SIG1(I,J) = SIG1(I,J) + PHI(II) * SIG(II,JJ)         00005950
   JT = JB + 1                                            00005960
320 SIG1(I,J) = SIG1(I,J) / PH(I)                        00005970
   SIG1(I,I) = 0.0                                         00005980
310 IT = IB + 1                                           00005990
                                                    00006000
   WRITE OUTPUT TAPEN6,350,(JIJ,JIJ=1,I6)                00006010
350 FORMAT(1H1///24X,14HFEW GROUP FLUX//I10,5(I14))      00006020
   WRITE OUTPUT TAPEN6,420,(PH(I),I=1,I6)                00006030
420 FORMAT(1PE16.7,1P5E15.7)                             00006040
   WRITE OUTPUT TAPEN6,351,(JIJ,JIJ=1,I6)                00006050
351 FORMAT(///18X,27HFEW GROUP SCATTERING MATRIX//27X,10HFROM - - -// 00006060
X: I23,5I15)                                              00006070
   WRITE OUTPUT TAPEN6,411                                00006080
411 FORMAT(//8H TO - -)                                   00006090
   DC 353 J=1,I6                                           00006100
   WRITE OUTPUT TAPEN6,352, J,(SIG1(I,J),I=1,I6)         00006110
352 FORMAT(I11,2X,1P6E15.7)                             00006120
353 CONTINUE                                              00006130
   IF(I6M ) 335,128,335                                    00006140
335 CALL PUNSH                                           00006150
128 RETURN                                               00006160
   END                                                    00006170
* LABEL(PUNSH )                                           00006180
  SUBROUTINE PUNSH                                         00006190
  DIMENSION                                              00006200
1 ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780), 00006210
2 FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40), 00006220
3 IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40), 00006230
4 STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6), 00006240
5 CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6), 00006250
6 C8(15),C9(15),IL01(5),SCHI(40)                        00006260
  COMMON                                                  00006270
1 IBLANK,N0G,NDSD,NDSDU,NTH,LEM,AW,VUZERG,B,CHIIND,AFN,JZD,JZU,MX, 00006280
2 DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST 00006290
3, DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,B0P,X7,Y7,Z7,R8,Z8,R9, 00006300
4 OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL01 00006310
5,N5,N6,N9,I6M,SCHI                                     00006320

```

DIMENSION C11(6),SIL0(8),SILP(24)	00006330
I6M1 = I6 - 1	00006340
GO TO(1,2,3),I6M	00006350
	00006360
C FAIM LIBRARY PUNCH	00006370
	00006380
	00006390
1 PUNCH 1420,(CSA(I),I=1,I6),(CVS6(I),I=1,I6),(CSTR(I),I=1,I6),((SIG	00006400
11(I,J),J=1,I6),I=1,I6M1)	00006410
1420 FORMAT (6E12.5)	00006420
GO TO 100	00006430
	00006440
C ULCER LIBRARY PUNCH	00006450
	00006460
2 NTH = IL01(1)	00006470
IF(CVS6(1)) 441,440,441	00006480
441 PUNCH 1420,(CVS6(I),I=1,I6)	00006490
440 PUNCH 1420,(CSA(I),I=1,I6)	00006500
PUNCH 1420,(CSTR(I),I=1,I6)	00006510
NDSO = I6 - 1	00006520
NDSU = I6-NTH	00006530
KK=0	00006540
DO 470 J=2,I6	00006550
JM1 = J-1	00006560
DO 470 I=1,JM1	00006570
IF(J - NDSO - I) 473,473,470	00006580
473 KK = KK + 1	00006590
C8(KK) = SIG1(I,J)	00006600
470 CONTINUE	00006610
PUNCH 1420,(C8(I),I=1,KK)	00006620
KK = 0	00006630
JM1 = I6-1	00006640
DO 474 J= NTH,JM1	00006650
JF1 = J+1	00006660
DO 474 I= JF1,I6	00006670
IF(I - NDSU - J) 480, 480, 474	00006680
480 KK = KK + 1	00006690
C9(KK) = SIG1(I,J)	00006700
474 CONTINUE	00006710
PUNCH 1420,(C9(I),I=1,KK)	00006720
GO TO 100	00006730
	00006740
C FLOCO PUNCH	00006750
	00006760
3 JJ = I6 - 1	00006770

DO 10 I=1,JJ	00006780
14 II = I + 1	00006790
SUMS = 0.0	00006800
DO 11 J = II,I6	00006810
11 SUMS = SIG1(I,J) + SUMS	00006820
10 C11(I) = C5(I) - (SUMS + CSA(I))	00006830
C11(I6) = C5(I6) - CSA(I6)	00006840
	00006850
JK = I6 + 2	00006860
JK3 = 3*JK	00006870
WRITE OUTPUT TAPE N6, 5	00006880
5 FORMAT(52H0THE FOLLOWING FLOCC CARDS HAVE BEEN PUNCHED - - - -//)	00006890
DO 20 I = 1,I6	00006900
DO 22 J=1,JJ	00006920
II = I-J	00006930
IF(II) 70,70,71	00006940
71 SIL0(J+3) = SIG1(II,I)	00006950
GO TO 22	00006960
70 SIL0(J+3) = 0.0	00006970
22 CONTINUE	00006980
7 SIL0(1) = CVS6(I)	00007000
SIL0(2) = C5(I)	00007010
SIL0(3) = C11(I)	00007020
CALL FLOPUN(SIL0(1),SILP(1),JK)	00007030
WRITE OUTPUT TAPE N6,24,(SILP(K),K=1,JK3)	00007040
24 FORMAT(6X,12A6)	00007050
20 CONTINUE	00007060
100 RETURN	00007070
END	00007080
SUBROUTINE FLOPUN (E,F,N)	00007090
DIMENSION E(2),F(2)	00007100
CALL FLOPN (E,F,N)	00007110
NN=3*N	00007120
DO 10 I=1,NN,12	00007130
II=I+11	00007140
IF(NN-II)1,2,2	00007150
1 II=NN	00007160
2 CONTINUE	00007170
PUNCH 5,(F(J),J=I,II)	00007180
5 FORMAT(12A6)	00007190
10 CONTINUE	00007200
RETURN	00007210
END	00007220
* FAP	00007230
COUNT 100	00007240

```

*      THIS IS FLOPN SUBROUTINE FOR W. RHOADES                                00007250
*****00007260
*      THIS ROUTINE IS TO MAKE AVAILABLE BCD FLOCO CARD IMAGES                00007270
*                                                                              00007280
*      AND CHANGE FLOATING POINT NUMBERS TO BCD                             00007290
*                                                                              00007300
*      EXECUTION TO BE CONTROLLED BY A FORTRAN STATEMENT                     00007310
*      SIMILAR TO                                                             00007320
*      DIMENSION A(100), B(300)                                              00007330
*      CALL FLOPN (A(5), B(13), N)                                           00007340
*****00007350
      EJECT
      ENTRY      FLOPN
KHALF OCT      000000606060
RHALF OCT      777777000000
TEN      OCT      000010000000
MINSE OCT      000040000025
PLUSE OCT      000020000025
PLUS     OCT      002000000000
MINUS    OCT      004000000000
K9CPT OCT      110033000000
KBCPT OCT      600033000000
K3CCT OCT      000000000003
CNTR     PZE
K1CCT OCT      000000000001
K9CDD OCT      110033000000
ZERO     OCT      000000000000
TAG1     PZE      0,1,0
TAG2     PZE      0,2,0
CHAR     OCT      233000000000
LHALF    OCT      606000000000
ONE      OCT      000001000000
B1       PZE
B2       PZE
B3       PZE
B4       PZE
B5       PZE
      D1 PZE
      D  PZE
      AK PZE
      BK PZE
      PZE
      PZE
      PZE
FLOPN    SXD      *-2,2

```

	SXD	*-2,1	00007700
	CLA	1,4	00007710
	STO	AK	00007720
	CLA	2,4	00007730
	STO	BK	00007740
	CLA*	3,4	00007750
	SUB	CNE	00007760
	STD	TEST	00007770
	CLA	TAG2	00007780
	STT	AK	00007790
	CLA	TAG1	00007800
	STT	BK	00007810
	SXD	BK+1,4	00007820
	AXT	0,2	00007830
	AXT	0,1	00007840
	STZ	CNTR	00007850
	CLA	K9CDD	00007860
	STO	K9DPT	00007870
BEGIN	CLA*	AK	00007880
	STO	D	00007890
	TSX	\$BINBCD,4	00007900
	PZE	D	00007910
C	PZE	B1	00007920
K	PZE	B2	00007930
	CLA	ZERO	00007940
	XCA		00007950
	LDQ*	C	00007960
	LGL	18	00007970
	SSP		00007980
	LDQ	D	00007990
	TQP	*+3	00008000
	ADD	MINUS	00008010
	TRA	*+2	00008020
	ADD	PLUS	00008030
	GRA	K9DPT	00008040
	SLW*	BK	00008050
	CLA	CNTR	00008060
	SUB	K3OCT	00008070
	TZE	MID1	00008080
	TRA	MID2	00008090
MID1	STZ	CNTR	00008100
	CLA	K9CDD	00008110
	STO	K9DPT	00008120
	TRA	MID3	00008130
MID2	CLA	CNTR	00008140

	ADD	K10CT	00008150
	ST0	CNTR	00008160
	CLA	KBCPT	00008170
	ST0	K9DPT	00008180
MID3	TXI	*+1,1,1	00008190
	LDQ*	C	00008200
	LGL	18	00008210
	CLA	ZERO	00008220
	LLS	35	00008230
	ST0	B5	00008240
	CAL	B5	00008250
	ANA	RHALF	00008260
	ORA	KHALF	00008270
	SLW*	BK	00008280
	TXI	*+1,1,1	00008290
	CLA*	K	00008300
	ARS	18	00008310
	ORA	CHAR	00008320
	FAD	CHAR	00008330
	ST0	D1	00008340
	TSX	BINCCD,4	00008350
	PZE	D1	00008360
C1	PZE	B3	00008370
K1	PZE	B4	00008380
	CLA	B2	00008390
	SSP		00008400
	SUB	TEN	00008410
	TPL	CN1	00008420
	CLA	ZERO	00008430
	LDQ*	C1	00008440
	LGL	6	00008450
	TRA	CN2	00008460
CN1	CLA	ZERO	00008470
	LDQ*	C1	00008480
	LGL	12	00008490
CN2	ALS	6	00008500
	LDQ*	K	00008510
	TQP	*+3	00008520
	ADD	MINSE	00008530
	TRA	*+2	00008540
	ADD	PLUSE	00008550
	ORA	LHALF	00008560
	SLW*	BK	00008570
	TXI	*+1,1,1	00008580
	TXI	*+1,2,1	00008590

```

TEST  TXL      BEGIN,2,0                                00008600
      LXD      FLCPN-3,4                                00008610
      LXD      FLCPN-2,2                                00008620
      LXD      FLCPN-1,1                                00008630
      TRA      4,4                                       00008640
      END                                             00008650
*      LABEL(PLCTC )                                00008999
      SUBROUTINE PLCTC                                00009000
      DIMENSION                                       00009010
1      ENG(40),CHI(40,40),SSF(40),SSA(40),SST(40),DNSTR(780),UPSTR(780),00009020
2      FCHI(40),IA(40),AA(40,40),BB(40,1),PHI(40),SIG(40,40),SS(40),    00009030
3      IC(6),B2(40),NAME(20),DI(20),TITLE(10),SF(40),SA(40),ST(40),    00009040
4      STOT(40),DNS(780),UPS(780),OMEGA(40),SIG1(40, 6),PH(6),IL(6),    00009050
5      CSA(6),CVS6(6),CSTR(6),C1(6),C2(6),C3(6),C4(6),C5(6),C6(6),C7(6),00009060
6      C8(15),C9(15),IL01(15),SCHI(40)                00009070
7      AEXX(9),OEXX(9),ALXX(9),OLXX(9),ENGG(40),U(40),XXX(80),YYY(80) 00009080
      COMMON                                           00009090
1      IBLANK,NOG,NDSD,NDSDU,NTH,LEM,AW,VUZERG,B,CHIIND,AFN,JZD,JZU,MX, 00009100
2      DII,IC,B2,NAME,DI,TITLE,SF,SA,ST,STOT,DNS,UPS,ENG,CHI,SSF,SSA,SST00009110
3      DNSTR,UPSTR,FCHI,IA,AA,BB,PHI,SIG,SS,N12,BOP,X7,Y7,Z7,R8,Z8,R9,    00009120
4      OMEGA,SIG1,PH,IL,CSA,CVS6,CSTR,C1,C2,C3,C4,C5,C6,C7,C8,C9,I6,IL0100009130
5      N5,N6,N9,I6M,SCHI,AEXX,OEXX,ALXX,OLXX,ENGG,U,XXX,YYY            00009140
C      PLOTS SPECTRUM GRAPHS                                00009150
      ETOP = 10.0E+06                                00009160
      ENGG(1)=ETOP -ENG(1)                             00009170
      U(1)= LOGEF(ETOP/ENGG(1))                       00009180
      SIG1(1,4) = PHI(1)/U(1)                         00009190
      DO 2001 I=2,N0G                                  00009200
2000  ENGG(I) = ENGG(I-1)-ENG(I)                       00009210
      U(I) = LOGEF(ETOP/ENGG(I))                     00009240
2001  SIG1(I,4)= PHI(I)/(U(I)-U(I-1))                 00009250
      IF(IC(6)) 2C04,2003,2005                        00009260
2004  J=0                                              00009270
      NUX = 2*N0G                                     00009272
      NUXM = NUX-2                                    00009276
      XXX(1) =0.0                                     00009280
      XXX(NUX) = U(N0G)                               00009290
      DO 2444 I=2,NUXM,2                             00009300
      J=J+1                                           00009310
      XXX(I) = U(J)                                   00009320
2444  XXX(I+1) =XXX(I)                               00009330
      J=0                                             00009340
      DO 2445 I=1,NUX ,2                             00009350

```

J=J+1	00009360
YYY(I) = SIG1(J,4)	00009370
2445 YYY(I+1) = YYY(I)	00009380
CALL AICRT 3(0,1,XXX(1),YYY(1),NUX,1,2,1,42,TITLE,ALXX,0LXX,3,1,16	00009390
1.0,16.0,1,XL,XU,1,YL,YU)	00009400
	00009410
2005 J=0	00009420
NNP1 =(N0G-NTH +1) * 2	00009425
NNPM = NNP1 - 2	00009426
XXX(NNP1) = ENGG(NTH-1)	00009430
XXX(1) = ENGG(N0G)	00009440
DO 444 I=2,NNPM,2	00009450
J=J+1	00009460
NJNJ = N0G - J	00009470
XXX(I) = ENGG(NJNJ)	00009480
444 XXX(I+1) = XXX(I)	00009490
J = -1	00009500
DO 11 I=1,NNP1,2	00009510
J=J+1	00009520
JJ = N0G - J	00009530
YYY(I) = SIG1(JJ,1)	00009540
11 YYY(I+1) = YYY(I)	00009550
CALL AICRT 3(1,1,XXX(1),YYY(1),NNP1,1,2,1,42,TITLE,AEXX,0EXX,3,1,	00009560
116.0,16.0,1,XL,XU,1,YL,YU)	00009570
2003 RETURN	00009580
END	00009590

FORTRAN WORK SHEET

DECK NO. Data PROGRAMMER M. A. Boling DATE 10-31-63 PAGE 1 OF 1 JOB NO. 7000-00
FOR PROGRAM NO. 9W-005 TITLE QUICKIE Sample Problems - Input BY DEPT. 721 GROUP 131

								IDENTIFICATION	
1	10	20	30	40	50	60	70	73	80
SAMPLE PROBLEM -	READ NEW ELEMENTS	-	COD	E	CALC. B2		+ 0 + 1 + 1 - 1 + 0 - 1		1 0
HZR	0 . 0 5	ZR	0 . 0 3	U235	0 . 0 0 1 0				2 0
	0 . 0		2 0 . 0		3 0 . 0		2 0 . 0		3 0
6	1 0	1 4	2 3						4 0
SECOND PROBLEM -	ADD NA TO MATERIALS						+ 0 + 2 - 1 + 3 + 0 + 1		5 0
						NA	0 . 0 0 0 5		6 0
THIRD PROBLEM -	USE B2 = 0 . 0	-	NOTE FEW GROUP DIS	0 . 0			+ 0 + 2 + 0 + 1 + 0 + 0		7 0
	0 . 0								8 0
FOURTH PROBLEM -	USE B2 = 1 . 0	E - 2 0					+ 0 + 2 + 0 + 0 + 0 + 0		9 0
FIFTH PROBLEM -	WEIGHT B OVER OLD FLUX						+ 1 + 2 + 1 + 3 + 0 + 0		1 0 0
B	1 . 0								1 1 0
SIXTH PROBLEM -	BASE CASE FOR REFLECTOR	CALC. - CHI	OPTION				+ 3 + 1 + 2 + 2 + 1 + 0		1 2 0
U235	0 . 0 0 1 1 6 6	U238	0 . 0 0 0 0 8 4 6 2	HZR	0 . 0 5 4 5 4	ZR	0 . 0 2 9 1 7		1 3 0
NA	0 . 0 0 0 5 2 8	K	0 . 0 0 1 1 4 2	PU241	0 . 0 0 0 2				1 4 0
	0 . 0 0 7 2		0 . 0 4 7		0 . 0 3 2		0 . 0 2 0		1 5 0
	0 . 0 1 0 5		0 . 0 0 4 9		0 . 0 0 4		0 . 0 0 7 3		1 6 0
	0 . 0 0 8 2		0 . 0 0 7 0		0 . 0 0 6 1		0 . 0 0 6 7		1 7 0
	0 . 0 0 8		0 . 0 0 6 7		0 . 0 0 1 4		- 0 . 0 2 7 9		1 8 0
	- 0 . 0 2 7 9		- 0 . 0 2 7 9		- 0 . 0 2 7 9		- 0 . 0 2 7 9		1 9 0
	- 0 . 0 2 7 9		- 0 . 0 2 7 9		- 0 . 0 2 7 9				2 0 0
1	1 4	2 3							2 1 0
SEVENTH PROBLEM -	REFLECTOR CALC. -	ADD BE	AND DELETE	OTHERS			+ 2 + 2 + 1 + 0 + 1 + 0		2 2 0
BE	0 . 1 2 3								2 3 0

SAMPLE PROBLEM -READ NEW ELEMENTS - CODE CALC. B2

SAMPLE PROBLEM OUTPUT

1	HZR	4.9999999E-02
2	ZR	3.0000000E-02
3	U235	9.9999999E-04
4		-0.
5		0.
6		0.
7		0.
8		0.
9		0.
10		0.
11		0.
12		0.
13		0.
14		0.
15		0.
16		0.
17		0.
18		0.
19		0.
20		0.

IC EDIT--- 0 1 1 -1 0 -1

MX= 1 SOLUTION WAS SUCCESSFUL

I	PHI	PHI / DELTA E	D	NU SIGMA F	SIGMA A	DB2	SIGMA TOTAL
1	1.1563397E 00	1.6519138E-07	2.9563932E 00	3.5569999E-03	1.3499999E-03	7.0978755E-02	1.7641875E-01
2	2.1280724E 00	1.3300452E-06	2.2988506E 00	3.1959999E-03	1.4500000E-03	6.5423764E-02	1.9165376E-01
3	1.2589950E 00	2.5179899E-06	1.7935612E 00	3.0869999E-03	1.5399999E-03	5.8813483E-02	2.3512347E-01
4	1.7593551E 00	3.5187101E-06	1.1799410E 00	2.9879999E-03	1.7100000E-03	4.6667318E-02	2.3660730E-01
5	1.9130787E 00	6.3769289E-06	8.9919972E-01	3.5179999E-03	2.1099999E-03	3.9001218E-02	2.4924121E-01
6	1.1130454E 00	1.3410185E-05	6.5397947E-01	6.1249999E-03	3.5999999E-03	3.0863339E-02	4.0252832E-01
7	8.3977650E-01	5.9984035E-05	5.9426181E-01	1.0290000E-02	5.9499999E-03	2.8644069E-02	4.8858405E-01
8	7.2839935E-01	2.9730586E-04	6.0500461E-01	1.9359999E-02	1.1600000E-02	2.9050724E-02	5.2492071E-01
9	6.6394476E-01	1.4754328E-03	5.9392298E-01	4.5814999E-02	2.7849999E-02	2.8631189E-02	5.3931117E-01
10	4.1650527E-01	5.9500752E-03	5.6858565E-01	9.3099998E-02	5.8849999E-02	2.7658620E-02	6.7210858E-01
11	3.3437777E-01	1.6718888E-02	5.3898187E-01	1.3475000E-01	9.1149998E-02	2.6498348E-02	7.2690330E-01
12	3.2250324E-01	4.6071891E-02	5.3806834E-01	1.0780000E-01	8.5699997E-02	2.6462128E-02	7.0749207E-01
13	2.8027039E-01	1.4013519E-01	5.7955896E-01	7.3499998E-02	4.1909999E-02	2.8082122E-02	6.9045208E-01
14	2.0864052E-01	3.4773420E-01	5.4024853E-01	1.7150000E-01	8.4579998E-02	2.6548529E-02	7.8135849E-01
15	1.3017261E-01	6.5086306E-01	3.2165048E-01	4.1265279E-01	2.1428909E-01	1.7127110E-02	9.6583511E-01
16	6.9521466E-02	1.1586911E 00	2.7225660E-01	4.6017539E-01	2.3475699E-01	1.4769360E-02	9.1365486E-01
17	1.4703961E-01	3.6759902E 00	2.4340050E-01	5.4618878E-01	2.7286479E-01	1.3349469E-02	3.3223744E-01
18	1.1813572E-01	3.9378574E 00	2.1469346E-01	6.8513489E-01	3.3986389E-01	1.1904836E-02	3.6125033E-01
19	1.0465957E-01	3.4886523E 00	1.8214837E-01	8.9524359E-01	4.4373028E-01	1.0227327E-02	4.6667832E-01
20	5.5212663E-02	2.7606332E 00	1.3996055E-01	1.2701462E 00	6.2883908E-01	7.9879987E-03	6.5408200E-01
21	1.9120094E-02	1.9120094E 00	9.9863039E-02	1.8741057E 00	9.2679697E-01	5.7894622E-03	9.5794371E-01
22	6.9695062E-03	1.1615844E 00	7.0426892E-02	2.8411503E 00	1.4050952E 00	4.1304821E-03	1.4474212E 00
23	1.6234376E-03	5.4114587E-01	4.5281229E-02	4.9825414E 00	2.4616641E 00	2.6822588E-03	2.5303246E 00

I	OMEGA	B2	CHI	FISSION SPECTRUM	ABSORPTION SPECTRUM	LEAKAGE SPECTRUM
1	6.3012506E 00	2.4008563E-02	2.0400000E-01	5.8758572E-10	2.2300836E-10	1.1725078E-08
2	4.8997656E 00	2.8459337E-02	3.4400000E-01	4.2508244E-09	1.9285655E-09	8.7016565E-08
3	3.8227926E 00	3.2791457E-02	1.6800000E-01	7.7730346E-09	3.8777043E-09	1.4809175E-07
4	2.5149239E 00	3.9550551E-02	1.8000000E-01	1.0513906E-08	6.0169942E-09	1.6420877E-07
5	1.9165525E 00	4.3373254E-02	8.9999999E-02	2.2434035E-08	1.3455320E-08	2.4870799E-07
6	1.3938905E 00	4.7193131E-02	1.4000000E-02	8.2137383E-08	4.8276665E-08	4.1388309E-07
7	1.2666084E 00	4.8201092E-02	-0.	6.1723571E-07	3.5690500E-07	1.7181869E-06
8	1.2895056E 00	4.8017359E-02	-0.	5.7558412E-06	3.4487478E-06	8.6369503E-06
9	1.2658863E 00	4.8206905E-02	-0.	6.7596951E-05	4.1090802E-05	4.2243395E-05
10	1.2118823E 00	4.8644597E-02	-0.	5.5395199E-04	3.5016192E-04	1.6457087E-04
11	1.1487849E 00	4.9163710E-02	-0.	2.2528701E-03	1.5239266E-03	4.4302293E-04
12	1.1468378E 00	4.9179865E-02	-0.	4.9665497E-03	3.9483609E-03	1.2191603E-03
13	1.2352708E 00	4.8454296E-02	-0.	1.0299936E-02	5.8730658E-03	3.9352935E-03
14	1.1514846E 00	4.9141326E-02	-0.	5.9636414E-02	2.9411358E-02	9.2318314E-03
15	6.8556518E-01	5.3247583E-02	-0.	2.6858045E-01	1.3947285E-01	1.1147403E-02
16	5.8028716E-01	5.4247941E-02	-0.	5.3320113E-01	2.7201084E-01	1.7113126E-02
17	5.1878334E-01	5.4845691E-02	-0.	2.0077846E 00	1.0030483E 00	4.9072517E-02
18	4.5759720E-01	5.5450390E-02	-0.	2.6979635E 00	1.3383356E 00	4.6879547E-02
19	3.8823067E-01	5.6148333E-02	-0.	3.1231936E 00	1.5480207E 00	3.5679589E-02
20	2.9831163E-01	5.7073218E-02	-0.	3.5064076E 00	1.7359940E 00	2.2051934E-02
21	2.1284788E-01	5.7974023E-02	-0.	3.5833077E 00	1.7720445E 00	1.1069506E-02
22	1.5010774E-01	5.8649218E-02	-0.	3.3002357E 00	1.6321367E 00	4.7979034E-03
23	9.6512319E-02	5.9235558E-02	-0.	2.6962817E 00	1.3321194E 00	1.4514933E-03

K EFF = 7.3625457E-01

FEW GROUP LIMITS MULTIGROUPS 6 10 14 23

GROUP- - - -	1	2	3	4
FISSION	3.36055E-02	9.19384E-02	1.36205E-01	4.74506E-01
ABSORPTION	1.76377E-02	5.64483E-02	8.75100E-02	2.37157E-01
LEAKAGE	4.86417E-01	7.57446E-02	3.08043E-02	8.28082E-03
D	1.45330E 00	5.93046E-01	5.48772E-01	2.30285E-01
NV SIGMA F	3.60231E-03	3.47117E-02	1.18874E-01	7.27262E-01
SIGMA A	1.89065E-03	2.13123E-02	7.63751E-02	3.63485E-01
SIGMA TR	2.62244E-01	5.62244E-01	6.07890E-01	1.58875E 00
D B2	5.21410E-02	2.85977E-02	2.68847E-02	1.26918E-02
B 2	3.58776E-02	4.82218E-02	4.89907E-02	5.51133E-02
1/(3*D)	2.29363E-01	5.62070E-01	6.07417E-01	1.44748E 00

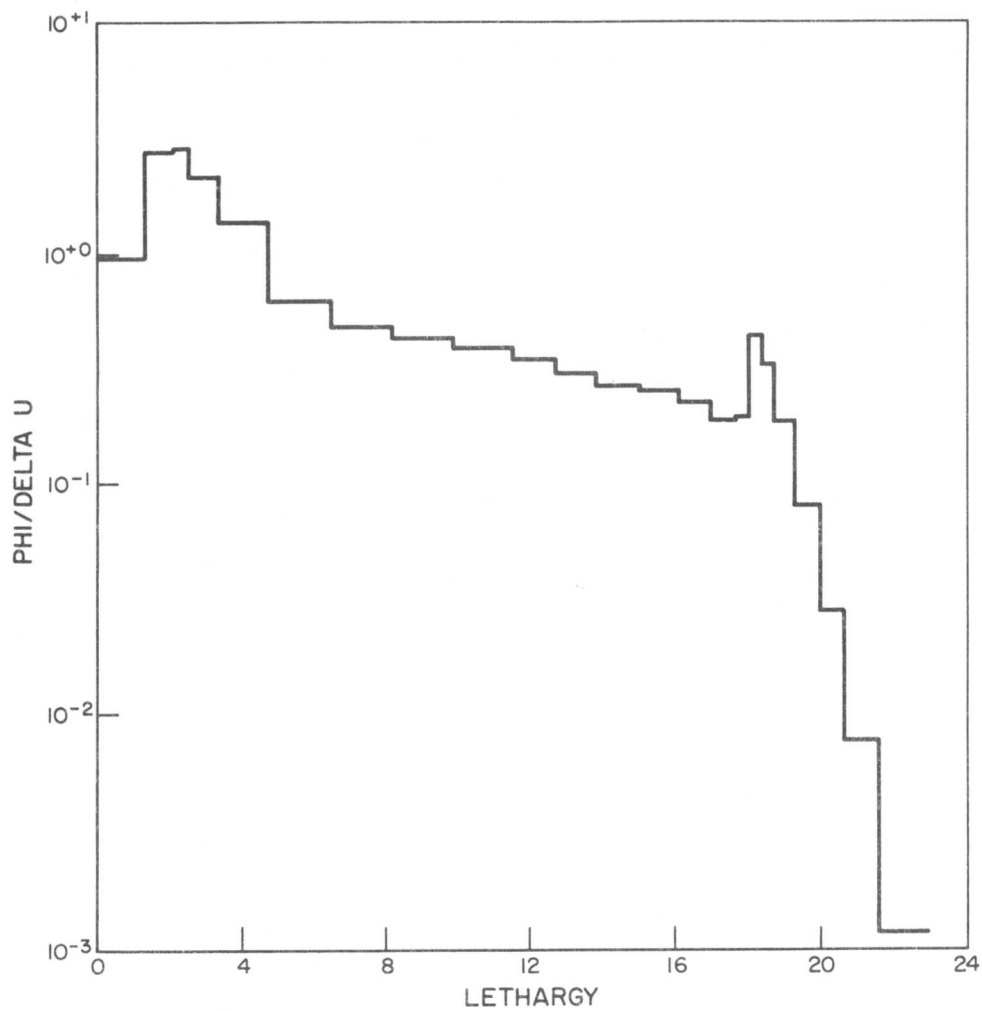
FEW GROUP FLUX

1	2	3	4
9.3288860E 00	2.6486259E 00	1.1457919E 00	6.5245466E-01

FEW GROUP SCATTERING MATRIX

FROM - - -

	1	2	3	4
TO - -				
1	0.	0.	0.	0.
2	5.3087758E-02	0.	0.	0.
3	7.4569820E-05	1.3584783E-01	0.	0.
4	0.	1.2257897E-03	2.1137476E-01	0.



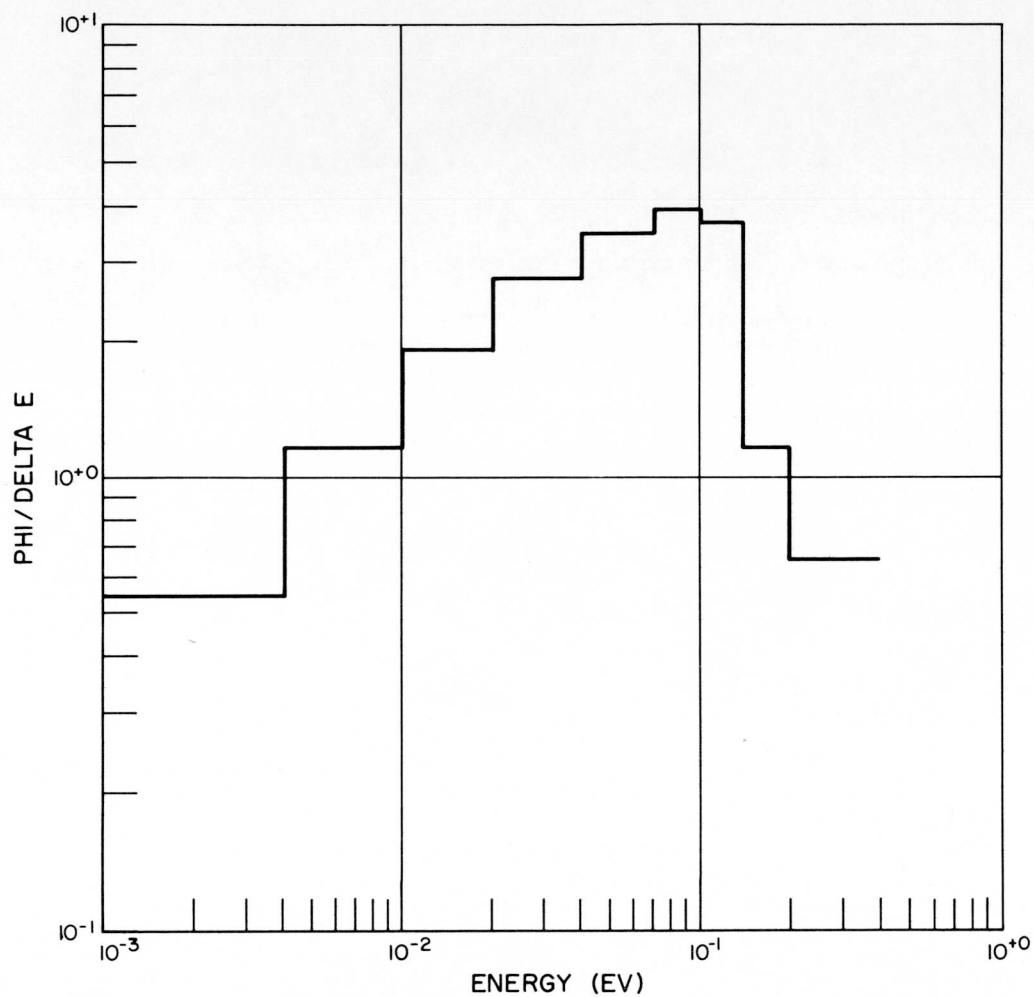
12-9-63

7569-01035

12-9-63

7569-01035

Sample Problem — Read New Elements — Code Calc.



12-9-63

7569-01047

Sample Problem — Read New Elements — Code Calc.