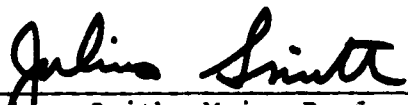


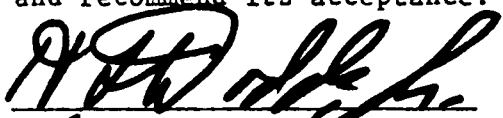
To the Graduate Council:

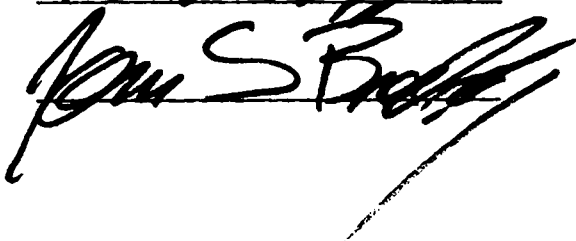
I am submitting herewith a thesis written by Jay Manneschmidt entitled "Coupling the DOT-IV Two-Dimensional Discrete Ordinates Transport Computer Code Through a 90-Degree Rotation of Geometry." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mathematics.



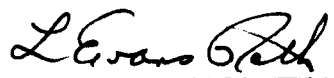
Julius Smith, Major Professor

We have read this thesis
and recommend its acceptance:





Accepted for the Council:



Vice Chancellor
Graduate Studies and Research

COUPLING THE DOT-IV TWO-DIMENSIONAL DISCRETE ORDINATES
TRANSPORT COMPUTER CODE THROUGH A 90-DEGREE ROTATION OF GEOMETRY

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Jay Mannes Schmidt

March 1982

3058601

ACKNOWLEDGMENTS

I wish to thank M. L. Williams for providing the advice, many theoretical notes, and previously used codes which got me started on this project. I offer special thanks to M. B. Emmett and W. A. Rhoades for their constant expert assistance and advice while debugging and testing the coupling program, and to D. A. Reed, a nuclear engineering student, for clarifying certain concepts and conventions which were unfamiliar to me. I also wish to thank H. L. Dodds for his guidance in coordinating this project in general, and his many suggestions for the improvement of this report. Finally, I wish to express appreciation to Julius Smith for his skillful and generous mathematical assistance, and the Mathematics and Nuclear Engineering Department heads at the University of Tennessee for arranging the research assistantship that made this work possible.

This work was supported by the Engineering Physics Division and Computer Sciences at Oak Ridge National Laboratory which is operated by the Union Carbide Corporation Nuclear Division for the U. S. Department of Energy under contract number W-7405-eng-26.

ABSTRACT

COUPL is a FORTRAN computer code designed to couple together two DOT-IV¹ discrete ordinates transport problems through a region where the r - z geometry is rotated 90 degrees. A typical case of this situation is the test problem in this report as depicted in Fig. 1 where two DOT-IV calculations are coupled through a 90-degree piping elbow (two-legged duct). COUPL is applicable to other problems of this type as long as the diameter of leg 2 is less than or equal to the diameter of leg 1.

COUPL reads the solutions calculated by the DOT-IV code in the region of interest of leg 1 and writes a boundary source, consisting of directional fluxes, on a specified r - θ (z boundary) surface in leg 2. This boundary source serves to start the leg 2 DOT-IV calculation and thus continue the solution of the piping elbow transport problem by producing reasonable solutions throughout leg 2.

After a brief discussion of the nature of the transport problem and the application of the DOT-IV code, and after outlining the requirements for the DOT-IV boundary source and characteristics of the DOT-IV VARFLM flux and moment output, a description of the transformation techniques as used in COUPL will follow. Finally, the sample problem (Fig. 1) on which the code has been tested will be discussed and the corresponding results will be compared with a Monte Carlo estimate. The comparison indicates that COUPL is a convenient and reasonably accurate means of effecting such a rotation in geometry in a transport problem, and is considerably less expensive than performing the same calculation by Monte Carlo.

ORNL-DWG 81-17711

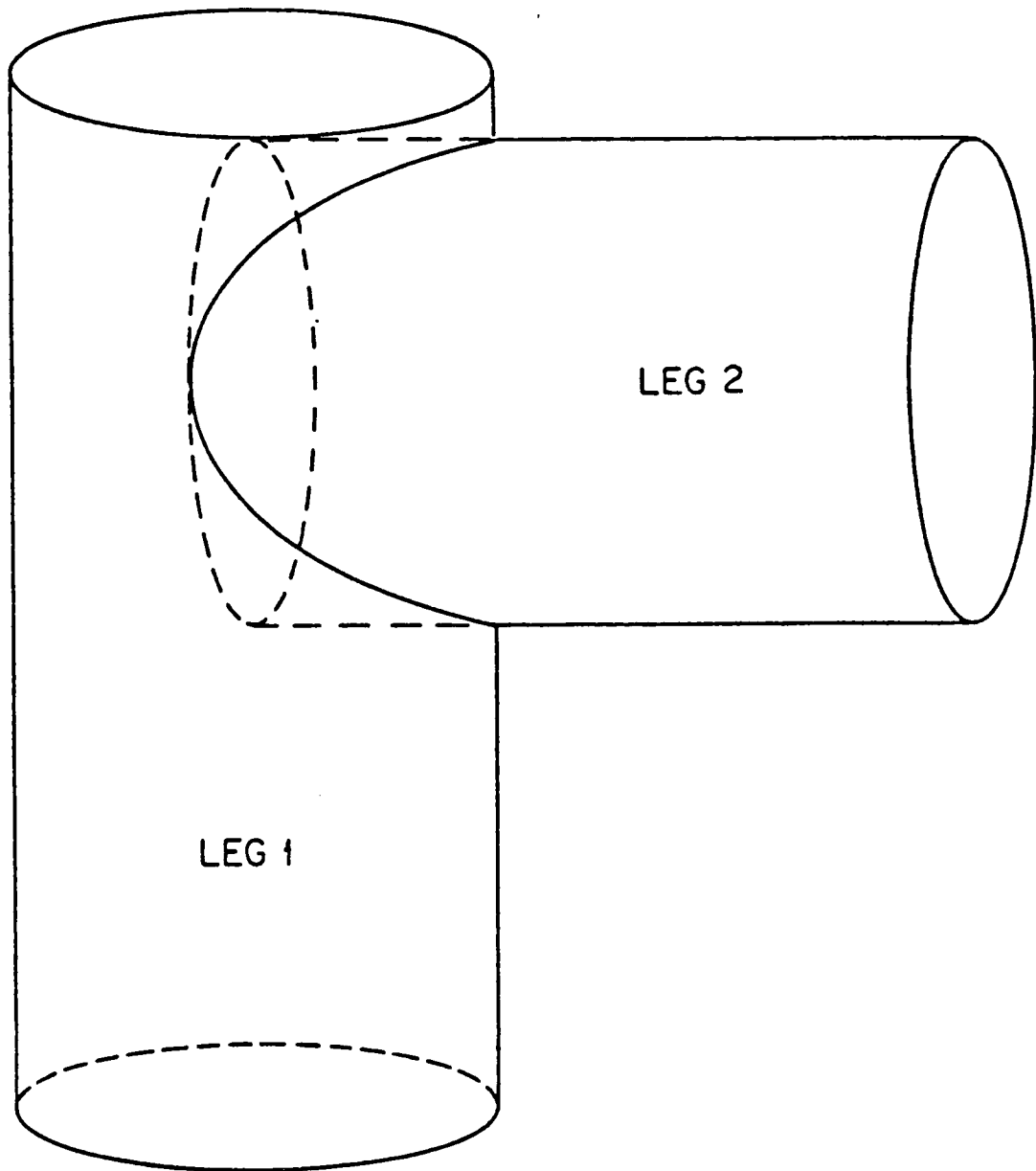


Fig. 1. 90 degree Piping Elbow Problem.

TABLE OF CONTENTS

SECTION	PAGE
I. INTRODUCTION	1
A. The Boltzmann Transport Equation.	1
B. Methods of Numerical Solution	4
II. THE DOT-IV TWO-DIMENSIONAL DISCRETE ORDINATES TRANSPORT CODE.	5
A. Introduction	5
B. Input Requirements.	8
C. Flux Moments Solution Format	16
III. STATEMENT OF THESIS PROBLEM	28
IV. COUPL--COUPLING DOT-IV CALCULATIONS THROUGH A 90- DEGREE ROTATION OF THE r - z COORDINATE SYSTEM.	30
V. NUMERICAL TESTING: RESULTS AND COMPARISONS	40
VI. CONCLUSIONS	50
LIST OF REFERENCES	51
APPENDICES	53
A. FIDO INPUT.	54
B. DOT-IV CARD INPUT SPECIFICATIONS.	61
C. GIP INPUT CARD SPECIFICATIONS	82
D. COUPL INPUT CARD SPECIFICATIONS	84
E. PROGRAM ACCESS.	86
VITA.	99

LIST OF FIGURES

FIGURE	PAGE
1. Unit Vectors $\bar{\Omega}$ and $\bar{\Omega}'$	3
2. Typical DOT-IV Transport Problem In Cylindrical Geometry	6
3. Coordinate System for Cylindrical Geometry showing Two-Dimensional Direction Vector $\bar{\Omega}=(\eta,\mu)$	7
4. Quadrature Set 'Igloo'	10
5. BNDRYS File Structure	13
6. VARFLM File Structure	24
7. Intersecting Perpendicular Cylinders	29
8. Mesh Embedded Boundary Source.	31
9. 'Bootstrapping' Technique	32
10. Transforming Position.	34
11. Transforming Direction	37
12. COUPL Test Problem	41
13. Isoplot--Leg 1	42
14. Isoplot--Leg 2	43
15. Scalar Flux Attenuation Curves With RDC=25 cm: Leg 1 and Leg 2	46
16. A Modeling Problem	48
A-1. Sample Data Sheet.	58
B-1. Triangular Geometry Options.	75

I. INTRODUCTION

A. The Boltzmann Transport Equation

The basic integro-differential equation used to describe neutron migration and multiplication is the steady state Boltzmann transport equation:

$$\begin{aligned}
 & \bar{\Omega} \cdot \nabla \psi(\bar{r}, \bar{\Omega}, E) + \sigma_t(\bar{r}, E) \psi(\bar{r}, \bar{\Omega}, E) \\
 &= \int_{E'} \int_{\bar{\Omega}'} \sigma_s(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}, E' \rightarrow E) \psi(\bar{r}, \bar{\Omega}', E') d\bar{\Omega}' dE' \\
 &+ \chi(E) \int_{E'} \int_{\bar{\Omega}'} v \sigma_f(\bar{r}, E') \psi(\bar{r}, \bar{\Omega}', E') d\bar{\Omega}' dE' + Q(\bar{r}, \bar{\Omega}, E) \quad .^2 \quad (1)
 \end{aligned}$$

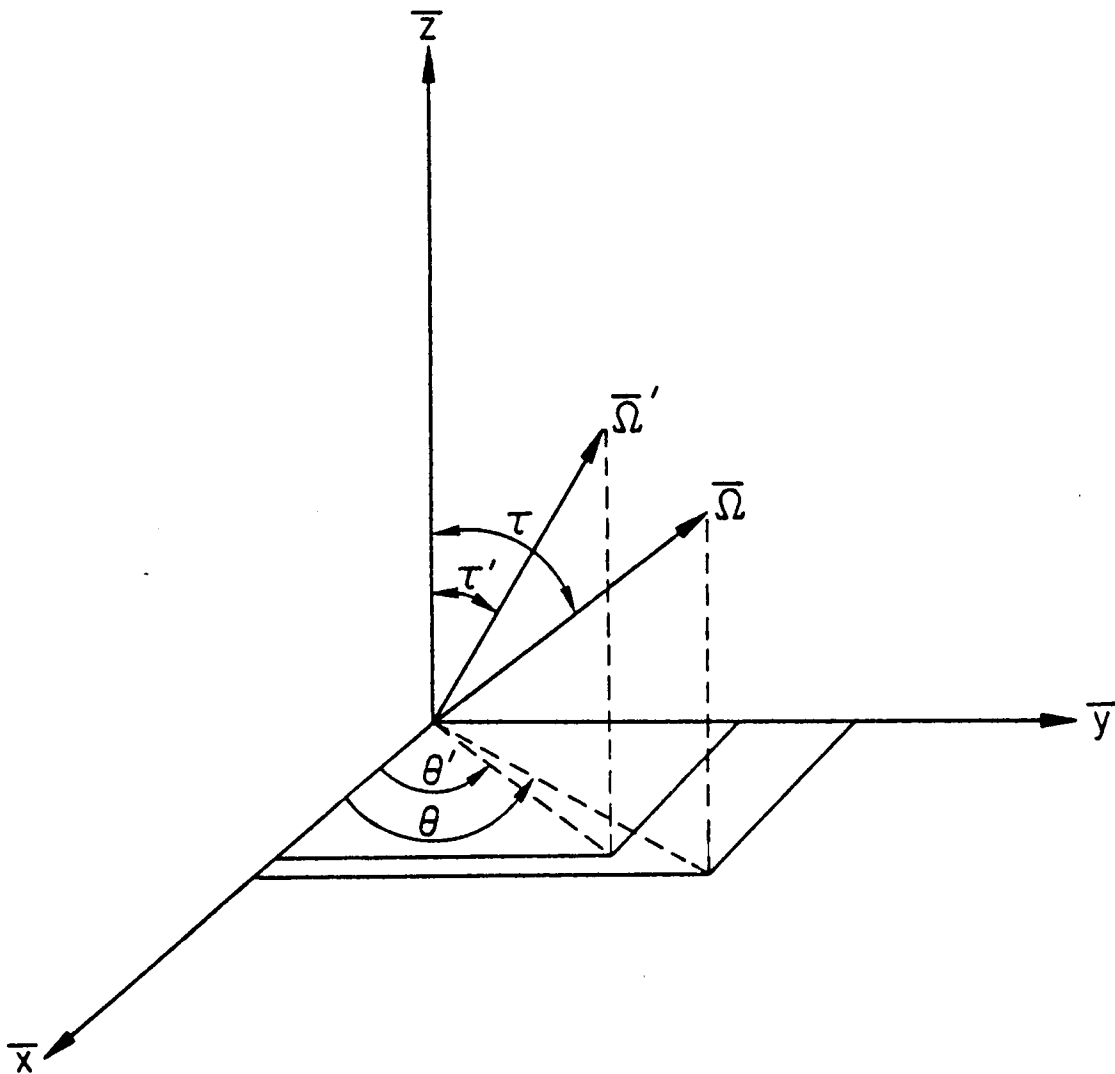
The equation describes the motion of neutral particles (i.e., neutrons and gammas) which travel in straight lines with constant speeds between collisions within a void, but as they pass through a material medium are subject to a certain probability of collision with the nuclei of the material. The effect of all of these collisions upon the propagating neutron population depends upon the respective type of each individual collision. A collision may result in the absorption of the colliding neutron by the target nucleus, thus removing it from the population, or the neutron may be 'scattered' to another direction and/or energy level. In a fissionable medium some collisions may cause the target nuclei to fission, and thus produce new particles.

The solution to Eq. (1), $\psi = \psi(\bar{r}, \bar{\Omega}, E)$, is the directional flux which is expressed in units of number of neutrons per unit area, per second,

per unit solid angle per unit energy. ψ is a function of position, $\bar{r}$, and velocity, $\bar{v}$, where $\bar{\Omega}$ is the unit direction vector in the velocity space and $E=E(\bar{v})$ is the energy. For two-dimensional problems, the unit direction vector, $\bar{\Omega} = (\tau, \theta)$, can be denoted as an ordered pair of angles as defined in Fig. 1.

Since Eq. (1) is time independent, i.e., steady state, it describes a time steady balance condition at a phase point, $(\bar{r}, \bar{\Omega}, E)$, where the left-hand side of the equation represents losses of neutrons and the right-hand side of the equation represents sources of neutrons. The first term on the left-hand side, $\bar{\Omega} \cdot \nabla \psi(\bar{r}, \bar{\Omega}, E)$, represents neutron flux losses due to leakage. The second term on the left-hand side, $\sigma_t(\bar{r}, E) \psi(\bar{r}, \bar{\Omega}, E)$, where $\sigma_t(\bar{r}, E)$ is the 'total macroscopic interaction cross section' (simply the probability per path length of a neutron being lost from the $(\bar{\Omega}, E)$ phase element due to collision), represents a loss due to absorption and scattering. The first term on the right-hand side, $\int_{E'} \int_{\bar{\Omega}'} \sigma_s(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}, E' \rightarrow E) \psi(\bar{r}, \bar{\Omega}', E') d\bar{\Omega}' dE'$, where $\sigma_s(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}, E' \rightarrow E)$ is the differential scattering cross section from energy E' and direction $\bar{\Omega}'$ into the phase element $(\bar{\Omega}, E)$, represents gains due to neutron inscatter from other energies and directions. The second term on the right-hand side, $\chi(E) \int_{E'} \int_{\bar{\Omega}'} \nu \sigma_f(\bar{r}, E') \psi(\bar{r}, \bar{\Omega}, E') d\bar{\Omega}' dE'$, where $\chi(E)$ is the fission spectrum of the material (the fraction of neutrons emitted in a fission which are at energy E) and where ν is the number of neutrons emitted per fission and $\sigma_f(\bar{r}, E')$ is the 'macroscopic cross section for fission' for neutrons of energy E' , represents gains due to fission. The final term represents an extraneous source.

ORNL- DWG 81-17712

Fig. 1. Unit Vectors $\bar{\Omega}$ and $\bar{\Omega}'$.

B. Methods of Numerical Solution

There are several methods available for solving this steady state transport equation such as spherical harmonics, Fourier transforms, discrete ordinates, and Monte Carlo methods. Among these the discrete ordinates approach is (for two-dimensional problems) a convenient and relatively economical method to use on modern, large, high-speed computers, and is the technique used by the DOT-IV computer code.

The DOT-IV solution of the transport equation (Eq. 1) is accomplished by splitting the energy spectrum into discrete groups and solving the resulting system of equations by an 'outer' iteration procedure. Each 'outer' iteration involves the repeated solution of an integro-differential equation in the remaining variables $\bar{r}$ and $\bar{\Omega}$. Each time the equation is solved it must be discretized in $\bar{\Omega}$ and $\bar{r}$ resulting in still finer sets of equations which are solved as finite difference equations. The iterations involved at this level are called 'inner' iterations. This system of two nested iterations is used by the DOT-IV computer code with certain sophisticated rebalance techniques applied within the inner iterations to accelerate convergence. These, as well as the differencing scheme to be used, can be user selected and adjusted to a certain extent.

While the inner workings of the DOT-IV code are of no critical interest to this work, certain requirements as well as consequences of the code must be noted.

II. THE DOT-IV TWO-DIMENSIONAL DISCRETE ORDINATES TRANSPORT CODE

A. Introduction

The DOT-IV code will solve the transport equation, (Eq. 1), for the case of large shielding problems with complex material geometry and, possibly, variable spatial mesh and directional quadrature. More importantly, DOT-IV allows for the solution of such a problem as a single unit eliminating the need to couple calculations together in a piecewise fashion whenever a change in material geometry is encountered which requires a change in spatial mesh or directional quadrature. Thus, much time and effort is saved which would otherwise have been spent running coupled segments of a single problem.

A typical radiation transport problem to be solved by the DOT-IV code is that of a straight cylinder in r - z geometry with 'j-boundary' ($z=\text{const.}$) flux source surface as shown in Fig. 2.

The directional notation in a two-dimensional radiation transport calculation can be defined by the directional cosines with respect to the geometric coordinate axes. For cylindrical coordinates the notation will be as shown in Fig. 3, where the direction vector $\bar{\Omega}$ in two dimensions is described by the directional cosine pair (η, μ) . Note that each direction $\bar{\Omega} = (\eta, \mu)$ denotes a pair of directions $\bar{\Omega}_+ = (\eta, \mu, +\xi)$ and $\bar{\Omega}_- = (\eta, \mu, -\xi)$ in three dimensions where

$$\xi = \pm \sqrt{1 - \eta^2 - \mu^2}.$$

This is a consequence of representing a three-dimensional spacial direction with two-dimensional directional cosines and implies a symmetry with respect to the third dimensional axis, θ . Therefore, the flux will be an even function of ξ .

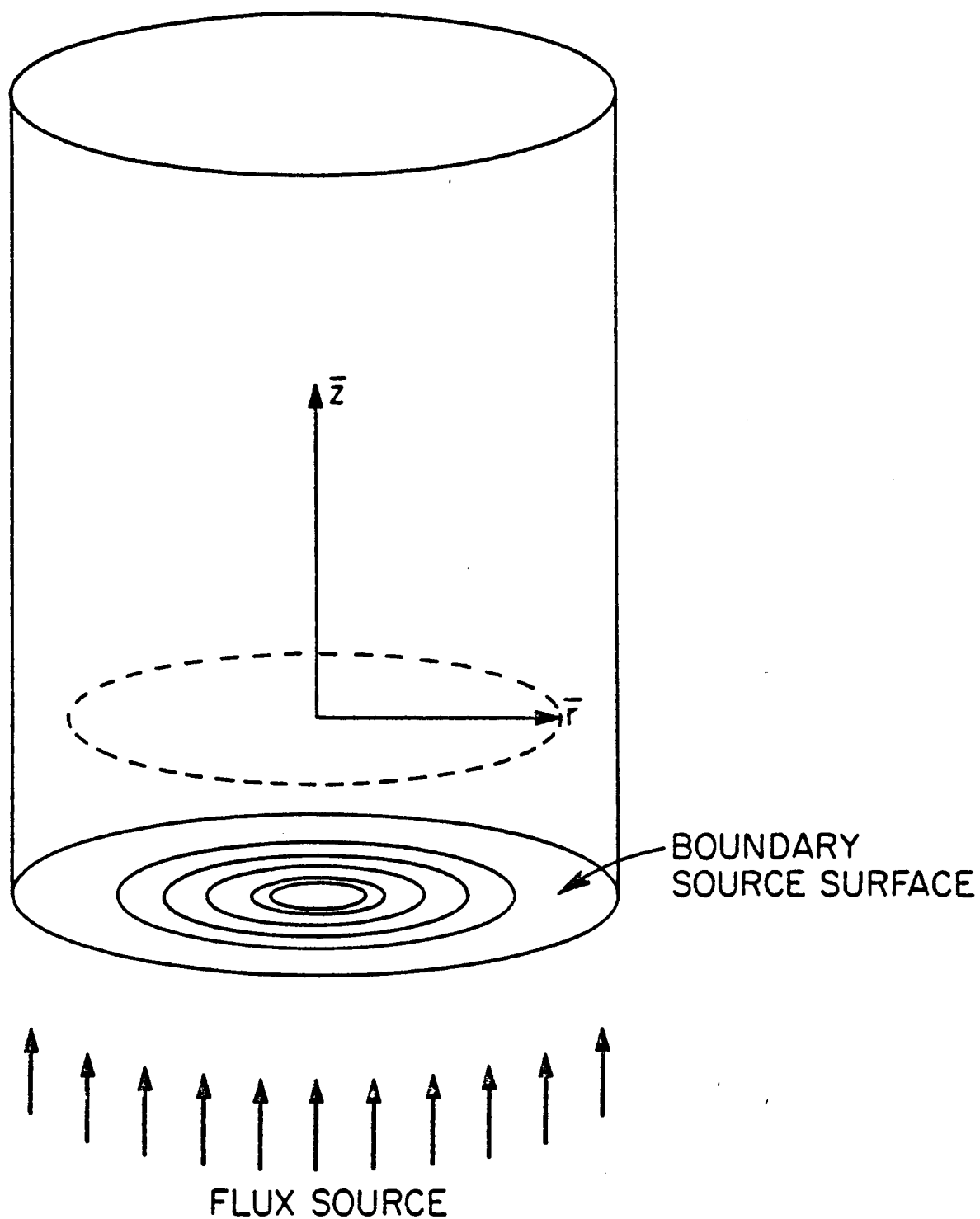


Fig. 2. Typical DOT-IV Transport Problem
In Cylindrical Geometry.

ORNL-DWG 81-17714

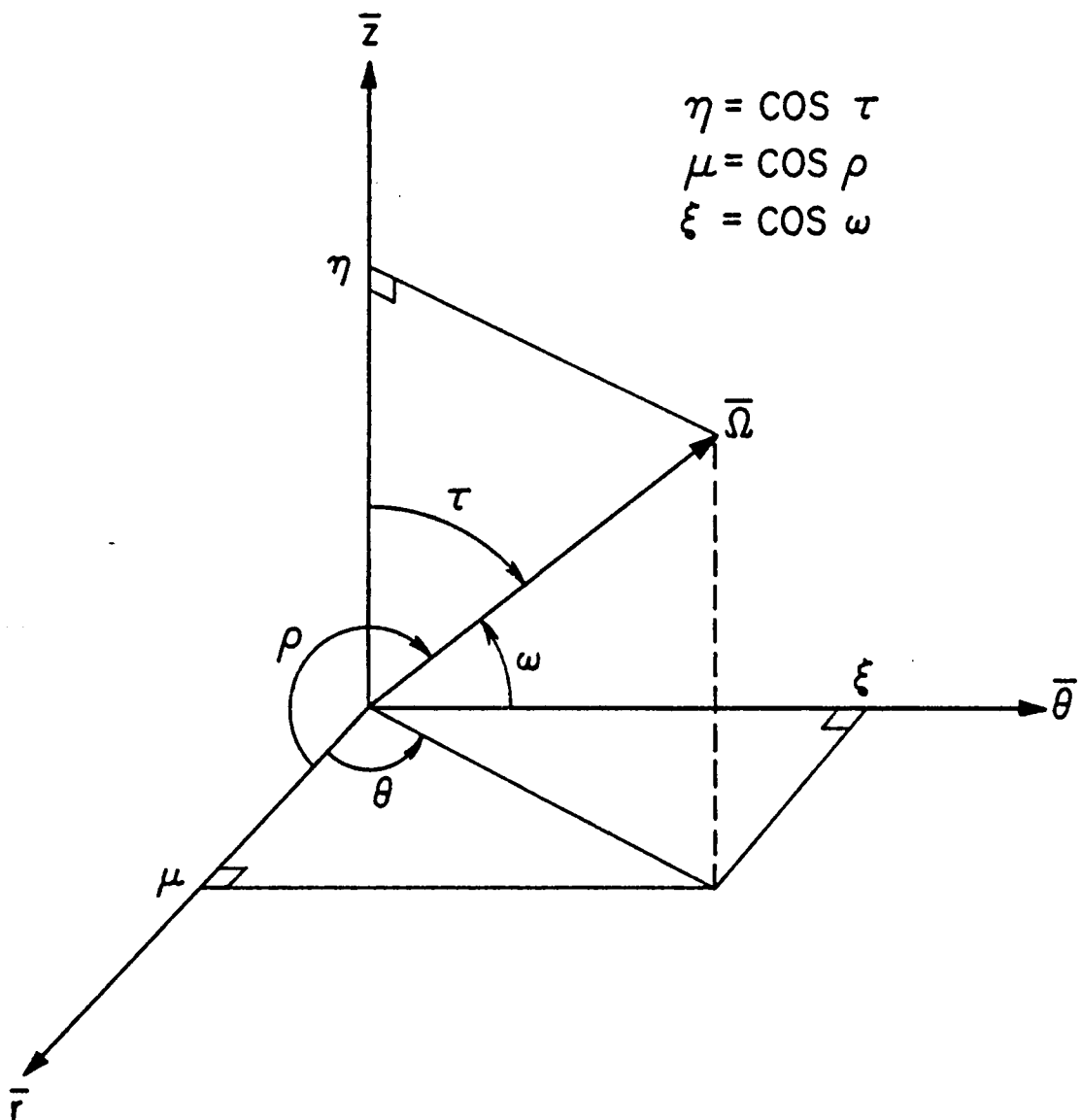


Fig. 3. Coordinate System for Cylindrical Geometry showing Two-Dimensional Direction Vector $\bar{\Omega} = (\eta, \mu)$.

In discrete ordinates calculations, such as DOT-IV, the angular variable, $\bar{\Omega}$, will be discretized into MM directional cosine pairs

$$\bar{\Omega}_m = (\eta_m, \mu_m), m=1, 2, \dots, MM.$$

Such a discrete set, usually selected in a particular way, is called a quadrature set and is of central importance later in this work.

B. Input Requirements

To specify the problem in Fig. 2, several parameters and boundary conditions must be provided to DOT-IV. The physical dimensions, spatial mesh, material zones, cross sections (absorption and scattering coefficients), and quadrature sets as well as the location of the source, must all be specified in addition to the source data itself.

The input for the DOT-IV code is read in using the FIDO input system. The FIDO input method is used by several major codes in the Engineering Physics and Computer Sciences departments at ORNL and is convenient for constructing large data arrays for input. An explanation of the FIDO system and its use is given in Appendix A, and the card input specifications for the DOT-IV code are given in Appendix B. Defining a spatial mesh for the cylindrical problem involves dividing the cylinder into a set of spatial mesh cells. For example, the z height of the cylinder is divided into JM intervals in widths which may vary. Typically, the interval width and density depends on the geometry and the material zones present in the problem. Similarly, the radius r is divided into IM variable-width intervals where IM may vary with z.

Material zones must be specified in the problem by designating a material number for every spatial mesh interval. The material numbers refer to corresponding cross-section coefficients necessary to solve

the Boltzmann equation through those intervals. The cross sections are read by DOT-IV from an external I/O device. For the DOT-IV calculations in this work the cross sections were processed and made available to DOT-IV by GIP. GIP is a group-organized cross-section input program which accepts nuclide organized microscopic cross-section data from card images in the input stream, or from the data library prepared by the ALC1 program. From this, GIP produces a group-organized file of microscopic and/or macroscopic cross sections for use by DOT-IV or other related codes. This includes the total scattering cross-section coefficients as well as the group-to-group inscatter cross-section coefficients. These are output on a specified I/O device to be read by DOT-IV. The input structure for GIP follows the FIDO input scheme (Appendix A) and is specified in Appendix C.

A quadrature set can be visualized as forming an 'igloo' with the discrete directions represented as vectors from the center of the floor to a point on the hemispherical shell (Fig. 4). The z axis points upward through the center of the 'igloo' and the r axis points outward at the floor level from the center of the igloo floor. A quadrature set may be grouped as follows:

$$\bar{\Omega}_{m(n,k)} = (n_n, \mu_{n,k}) \quad .$$

Note that each combination of n and k (n denotes polar angle level and k denotes azimuthal angle level) determines a unique direction vector $\bar{\Omega}_{m,m} = 1, 2, \dots, MM$. Associated with every quadrature set, there is a corresponding set of weights, $w_{n,k}$, which denote the fraction of the unit sphere ('igloo' shell) surface swept by the direction vector $\bar{\Omega}_{m(n,k)}$. (See Fig. 4.) Consequently we have,

$$\sum_n \sum_k w_{n,k} = 1 \quad .$$

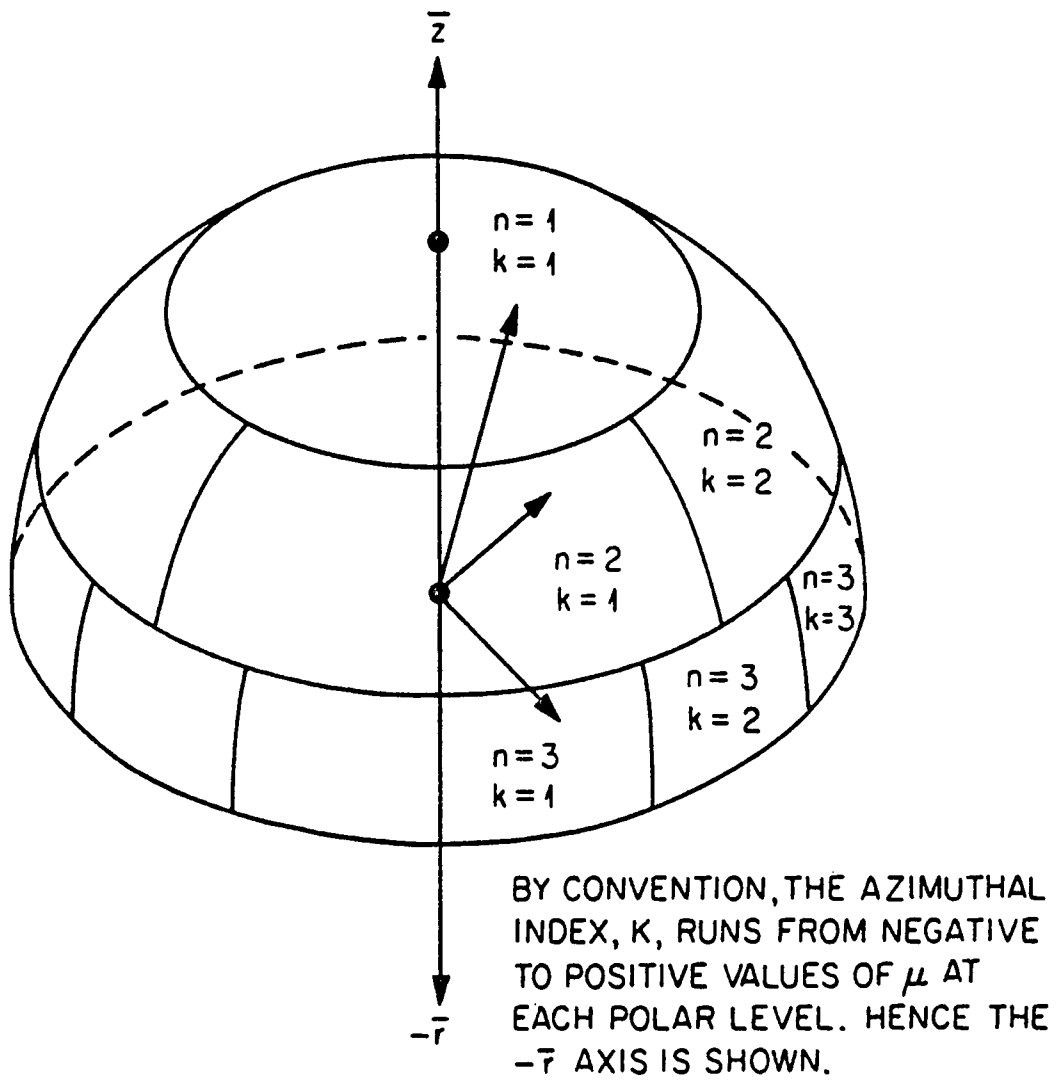


Fig. 4. Quadrature Set 'Igloo.'

A typical 48-direction quadrature set is given in Table 1.

Fully symmetric quadrature sets are those which exhibit complete rotational symmetry with respect to any coordinate axis. That is, given a set member $\bar{\Omega} = (\eta, \mu, \xi)$, the vector resulting from a 90-degree rotation of $\bar{\Omega}$ about any axis is also a member of the set. 'Half symmetrical' sets require such symmetry only about the third dimensional axis. 'Biased' sets are those quadrature sets which have an unequal number of directions in the positive or negative domains of one of the variables. Because they have a larger number of directions in some portion of the unit sphere, they may be useful in situations where neutron flow is expected to be focused about some known general direction.

The flux source for a DOT-IV calculation must be input according to a special format or file structure type. For the problem in Fig. 2 and the coupling problem later on in this work, an external boundary source is used, and uses the BNDRYS file structure format. The structure consists of four 'header' records followed by the actual source data records as shown in Fig. 5.

In a cylindrical problem, as in Fig. 2, the 'I-boundary directional source' can be characterized geometrically as a cylindrical shell surface of fixed r with a set of directional fluxes, corresponding to the directions in the chosen quadrature set, given at each z level on the cylindrical shell. Similarly, the 'J-boundary directional source' can be characterized as a 'disc' surface of fixed z with a set of directional fluxes, each set member corresponding to a direction, given at each r level on the disk surface. For the DOT-IV calculations in this work, only the J-boundary source is used. The other boundary surfaces are designated

Table 1. A Typical Direction Set (S-8).

	MU	S8 Summetric ETA	WEIGHT
1	-0.30861E 00	-0.95119E 00	0.0
2	-0.21822E 00	-0.95119E 00	0.30247E-01
3	0.21822E 00	-0.95119E 00	0.30247E-01
4	-0.61721E 00	-0.78680E 00	0.0
5	-0.57735E 00	-0.78680E 00	0.22685E-01
6	-0.21822E 00	-0.78680E 00	0.22685E-01
7	0.21822E 00	-0.78680E 00	0.22685E-01
8	0.57735E 00	-0.78680E 00	0.22685E-01
9	-0.81650E 00	-0.57735E 00	0.0
10	-0.78680E 00	-0.57735E 00	0.22685E-01
11	-0.57735E 00	-0.57735E 00	0.23148E-01
12	-0.21822E 00	-0.57735E 00	0.22685E-01
13	0.21822E 00	-0.57735E 00	0.22685E-01
14	0.57735E 00	-0.57735E 00	0.23148E-01
15	0.78680E 00	-0.57735E 00	0.22685E-01
16	-0.97590E 00	-0.21822E 00	0.0
17	-0.95119E 00	-0.21822E 00	0.30247E-01
18	-0.78680E 00	-0.21822E 00	0.22685E-01
19	-0.57735E 00	-0.21822E 00	0.22685E-01
20	-0.21822E 00	-0.21822E 00	0.30247E-01
21	0.21822E 00	-0.21822E 00	0.30247E-01
22	0.57735E 00	-0.21822E 00	0.22685E-01
23	0.78680E 00	-0.21822E 00	0.22685E-01
24	0.95119E 00	-0.21822E 00	0.30247E-01
25	-0.30861E 00	0.95119E 00	0.0
26	-0.21822E 00	0.95119E 00	0.30247E-01
27	0.21822E 00	0.95119E 00	0.30247E-01
28	-0.61721E 00	0.78680E 00	0.0
29	-0.57735E 00	0.78680E 00	0.22685E-01
30	-0.21822E 00	0.78680E 00	0.22685E-01
31	0.21822E 00	0.78680E 00	0.22685E-01
32	0.57735E 00	0.78680E 00	0.22685E-01
33	-0.81650E 00	0.57735E 00	0.0
34	-0.78680E 00	0.57735E 00	0.22685E-01
35	-0.57735E 00	0.57735E 00	0.23148E-01
36	-0.21822E 00	0.57735E 00	0.22685E-01
37	0.21822E 00	0.57735E 00	0.22685E-01
38	0.57735E 00	0.57735E 00	0.23148E-01
39	0.78680E 00	0.57735E 00	0.22685E-01
40	-0.97590E 00	0.21822E 00	0.0
41	-0.95119E 00	0.21822E 00	0.30247E-01
42	-0.78680E 00	0.21822E 00	0.22685E-01
43	-0.57735E 00	0.21822E 00	0.22685E-01
44	-0.21822E 00	0.21822E 00	0.30247E-01
45	0.21822E 00	0.21822E 00	0.30247E-01
46	0.57735E 00	0.21822E 00	0.22685E-01
47	0.78680E 00	0.21822E 00	0.22685E-01
48	0.95119E 00	0.21822E 00	0.30247E-01

```

C*****BND
C          PROPOSED  8 OCT 76          -BND
C          -BND
CF          BND RYS                    -BND
CE          MULTIPLE BOUNDARY SOURCE DATA -BND
C          -BND
C*****BND
CD          ORDER OF GROUPS IS BY DECREASING ENERGY BND
CD          I IS THE FIRST-DIMENSION INDEX          BND
CD          J IS THE SECOND-DIMENSION INDEX          BND
CD          JM=1 FOR 1-DIMENSIONAL GEOMETRY          BND
CD          NINTSR=NJNTSR=1 FOR EXTERNAL BOUNDARY DATA, BND
CD          AND Z(1)=R(1)=0 IN THAT CASE            BND
C-----BND
CS          FILE STRUCTURE                -BND
CS          -BND
CS          RECORD TYPE                    PRESENT IF -BND
CS          -----
CS          FILE IDENTIFICATION            ALWAYS      -BND
CS          FILE LABEL                     ALWAYS      -BND
CS          FILE CONTROL                   ALWAYS      -BND
CS          MESH DESCRIPTION               ALWAYS      -BND
CS          -BND
CS          ***** (REPEAT OVER ALL GROUPS) -BND
CS          *          I-BOUNDARY DIRECTIONAL SOURCE  NINTSR.GT.0 -BND
CS          *          J-BOUNDARY DIRECTIONAL SOURCE  NJNTSR.GT.0 -BND
CS          ***** -BND
C          -BND
C-----BND
C          BND
C          BND
C-----BND
CR          FILE IDENTIFICATION            -BND
C          -BND
CL          HNAME, (HUSE(I), I=1,2), IVERS -BND
C          -BND
CW          1+3*MULT=NUMBER OF WORDS      -BND
C          -BND
CD          HNAME          HOLLERITH FILE NAME - BND RYS - (A6) -BND
CD          HUSE(I)        HOLLERITH USER IDENTIFICATION - (A6) -BND
CD          IVERS          FILE VERSION NUMBER -BND
CD          MULT           DOUBLE PRECISION PARAMETER -BND
CD          1- A6 WORD IS SINGLE WORD -BND
CD          2- A6 WORD IS DOUBLE PRECISION WORD -BND
C          -BND
C-----BND
C          BND
C          BND

```

Fig. 5. BND RYS File Structure.

```

C-----BNDS
CR      FILE LABEL                                -BNDS
C                                             -BNDS
CL      DATE,USER,CHARGE,CASE,TIME,(TITL(I),I=1,12) -BNDS
C                                             -BNDS
CW      17*MULT=NUMBER OF WORDS                  -BNDS
C                                             -BNDS
CD      DATE          AS PROVIDED BY TIMER OPTION 4 - (A6) -BNDS
CD      USER          AS PROVIDED BY TIMER OPTION 5 - (A6) -BNDS
CD      CHARGE         AS PROVIDED BY TIMER OPTION 6 - (A6) -BNDS
CD      CASE           AS PROVIDED BY TIMER OPTION 7 - (A6) -BNDS
CD      TIME           AS PROVIDED BY TIMER OPTION 8 - (A6) -BNDS
CD      TITL(I)        TITLE PROVIDED BY USER      - (A6) -BNDS
C                                             -BNDS
C-----BNDS
C                                             BNDS
C                                             BNDS
C-----BNDS
CR      FILE CONTROL                                -BNDS
C                                             -BNDS
CL      IGM,JM,IMA,MMA,NINTSR,NJNTSR,(IDUM(I),I=1,19) -BNDS
C                                             -BNDS
CW      25=NUMBER OF WORDS                        -BNDS
C                                             -BNDS
CD      IGM            NUMBER OF ENERGY GROUPS      -BNDS
CD      JM             NUMBER OF SECOND-DIMENSION (J) INTERVALS -BNDS
CD      IMA            NUMBER OF FIRST-DIMENSION INTERVALS -BNDS
CD      MMA            NUMBER OF BOUNDARY DIRECTIONS -BNDS
CD      NINTSR         NUMBER OF I-BOUNDARY SOURCES -BNDS
CD      NJNTSR         NUMBER OF J-BOUNDARY SOURCES -BNDS
CD      IDUM(I)        ARRAY SET TO 0                -BNDS
C                                             -BNDS
C-----BNDS
C                                             BNDS
C                                             BNDS
C-----BNDS
CR      MESH DESCRIPTION                            -BNDS
C                                             -BNDS
CL      (Z(J),J=1,NJNTSR),(R(I),I=1,NINTSR)        -BNDS
C                                             -BNDS
CW      JM+IMA=NUMBER OF WORDS                      -BNDS
C                                             -BNDS
CD      Z(J)           POSITION FROM WHICH J-BOUNDARY SOURCES TAKEN -BNDS
CD      R(I)           POSITION FROM WHICH I-BOUNDARY SOURCES TAKEN -BNDS
C                                             -BNDS
C-----BNDS
C                                             BNDS
C                                             BNDS

```

Fig. 5. (continued).

C-----		-BNDS
CR	I-BOUNDARY DIRECTIONAL SOURCES	-BNDS
C		-BNDS
CL	(((BIJ(M,J,N),M=1,MMA),J=1,JM),N=1,NINTSR)	-BNDS
C		-BNDS
CW	MMA*JM*NINTSR=NUMBER OF WORDS	-BNDS
C		-BNDS
CD	BIJ(M,J,N) SOURCE IN DIRECTION M, INTERVAL J, SET N	-BNDS
C		-BNDS
C-----		-BNDS
		BNDS
		BNDS
C-----		-BNDS
CR	J-BOUNDARY DIRECTIONAL SOURCES	-BNDS
C		-BNDS
CL	(((BJI(M,I,N),M=1,MMA),I=1,IMA),N=1,NJNTSR)	-BNDS
C		-BNDS
CW	MMA*IMA*NJNTSR=NUMBER OF WORDS	-BNDS
C		-BNDS
CD	BJI(M,I,N) SOURCE IN DIRECTION M, INTERVAL I, SET N	-BNDS
C		-BNDS
C-----		-BNDS
		BNDS
		BNDS
		BNDS
		BNDS
END		BNDS

Fig. 5. (continued).

(in this work) as void, meaning that any neutron passing through the surface from the cylinder enters a vacuum (is lost) and cannot be scattered back into the population.

With these input requirements provided for, a DOT-IV calculation may be performed on the cylindrical problem in Fig. 2.

C. Flux Moments Solution Format

The discrete ordinates method involves the solution of a discretized set of equations corresponding to the Boltzmann equation, (Eq. 1), from which a resulting set of solution functions are calculated. They are:

$$\begin{aligned} \psi(\bar{r}_{ij}, \bar{\Omega}_m, E_g) \quad & i=1,2,\dots,IM \\ & j=1,2,\dots,JM \\ & m=1,2,\dots,MM \\ & g=1,2,\dots,IGM . \end{aligned}$$

They express the directional flux which is the flux magnitude in the spatial interval about (r_i, z_j) , in the direction $\bar{\Omega}_m$ and of the energy level E_g . This form of the solution involves $MM*IGM*IM*JM$ numbers and, therefore, is not very compact. A more compact form of the solution is the flux moment solution set. In addition, the flux moments provide a useful property for the coupling code in this work.

The flux moment set in the spatial interval about (i,j) and of energy level g is

$$\begin{aligned} \psi_{i,j,g}^n &= \psi_{\ell}^n(\bar{r}_{ij}, E_g) \quad \ell=1,2,\dots,OE \\ & \quad n=0,1,2,\dots,\ell \end{aligned}$$

where OE is the order of the moment expansion.

The moments within a set are the coefficients of corresponding spherical harmonics in an orthogonal series expansion of the directional flux, $\psi(\bar{r}, \bar{\Omega}, E)$. In general, the directional flux can be written:

$$\psi(\bar{r}, \bar{\Omega}, E) = \sum_{\ell=0}^{\infty} \sum_{n=-\ell}^{\ell} \psi_{\ell}^n(\bar{r}, E) Y_{\ell}^n(\bar{\Omega}) \quad , \quad (2)$$

where $Y_{\ell}^n(\bar{\Omega})$ is a spherical harmonic which will be defined shortly. The important thing to note here is that the directional flux (Eq. 2) is now, as in the analytical equation, expressed as a continuous function of the direction vector $\bar{\Omega}$. This is convenient because for any direction pair, (η, μ) , not necessarily a member of the quadrature set, one can obtain an approximation to the flux in that direction.

The flux moments are a consequence of the fact that one way of deriving the discrete ordinates finite difference transport equation involves the expansion, in spherical harmonics, of the continuous functions of $\bar{\Omega}$ in the source terms on the R.H.S. of Eq. 1. These are the inscatter integral,

$$\int_{E'} \int_{\bar{\Omega}'} \sigma_s(\bar{r}, \bar{\Omega}' \rightarrow \bar{\Omega}, E' \rightarrow E) \psi(\bar{r}, \bar{\Omega}', E') d\bar{\Omega}' dE' \quad , \quad (3)$$

and the fission source,

$$\chi(E) \int_{E'} \int_{\bar{\Omega}'} v \sigma_f(\bar{r}, E') \psi(\bar{r}, \bar{\Omega}', E') d\bar{\Omega}' dE' \quad . \quad (4)$$

In a shielding problem, such as in this work, there is generally no fission term. In any case, however, the functions to be expanded

are the directional flux, $\psi(\vec{r}, \vec{\Omega}, E)$, and the scattering cross section, $\sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E)$.

Spherical harmonics, $Y_\ell^n(\vec{\Omega})$, are a complete set of functions in the direction variables, (η, θ) , where $\eta = \cos(\tau)$ and θ is the azimuthal angle as shown in Fig. 3. They are constructed as follows:

$$Y_\ell^n(\vec{\Omega}) = Y_\ell^n(\eta, \theta) = \left\{ \frac{(2\ell+1)(\ell-n)!}{(\ell+n)!} \right\}^{1/2} P_\ell^n(\eta) \exp(in\theta) \quad , \quad 2 \quad (5)$$

where ℓ is a non-negative integer and $-\ell \leq n \leq \ell$. The $P_\ell^n(\eta)$ are the associated Legendre polynomials defined by:

$$P_\ell^n(\eta) = (1-\eta^2)^{\frac{n}{2}} \frac{d^n}{d\eta^n} P_\ell(\eta) \quad \text{and} \quad (6)$$

$$P_\ell^{-n} = (-1)^n \frac{(\ell-n)!}{(\ell+n)!} P_\ell^n(\eta) \quad . \quad (7)$$

Here, $P_\ell(\eta)$ are the Legendre polynomials, i.e.,

$$P_0(\eta) = 1, \quad P_1(\eta) = \eta$$

$$\text{and} \quad P_\ell(\eta) = \frac{2\ell-1}{\ell} \eta P_{\ell-1}(\eta) - (\ell-1)P_{\ell-2}(\eta) \quad , \quad (8)$$

for $\ell = 2, 3, \dots$

with the orthogonality property

$$\int_{-1}^1 \frac{2\ell+1}{2} P_\ell(\eta) P_\ell(\eta) d\eta = \delta_{r_\ell} \quad , \quad (8a)$$

where $\delta_{r,\ell}$ is the kronecker delta. From Eq. 8a we have

$$\int_{\bar{\Omega}} Y_r^s(\bar{\Omega}) \bar{Y}_\ell^n(\bar{\Omega}) d\bar{\Omega} = \frac{1}{4\pi} \int_{-1}^1 \int_0^{2\pi} Y_r^s(\eta, \theta) \bar{Y}_\ell^n(\eta, \theta) d\eta d\theta = \delta_{r\ell} \delta_{sn} \quad , \quad (9)$$

where $\bar{Y}_\ell^n(\bar{\Omega})$ is the complex conjugate of $Y_\ell^n(\bar{\Omega})$. Because the $Y_\ell^n(\bar{\Omega})$ form a complete set we may make the expansion in Eq. 2 and since the scattering cross section $\sigma_s(\bar{r}, \bar{\Omega} \rightarrow \bar{\Omega}, E' \rightarrow E)$ depends upon one angle, that is the scattering angle $\cos^{-1}(\bar{\Omega}' \cdot \bar{\Omega})$, we can let $\mu_0 = \bar{\Omega}' \cdot \bar{\Omega}$ be the cosine of the scattering angle and expand $\sigma_s(\bar{r}, \mu_0, E' \rightarrow E)$ in terms of Legendre polynomials, i.e.,

$$\sigma_s(\bar{r}, \mu_0, E' \rightarrow E) = \sum_{\ell=0}^{\infty} (2\ell+1) \sigma_{s\ell}(\bar{r}, E' \rightarrow E) P_\ell(\mu_0) \quad . \quad (10)$$

Orthogonality properties enable us to solve for the expansion coefficients ψ_ℓ^n and $\sigma_{s\ell}$ as follows:

$$\psi_\ell^n(\bar{r}, E') = \int_{\bar{\Omega}} Y_\ell^n(\bar{\Omega}) \psi(\bar{r}, \bar{\Omega}, E) d\bar{\Omega}$$

and

$$\sigma_{s\ell}(\bar{r}, E' \rightarrow E) = \frac{1}{2} \int_{-1}^1 P_\ell(\mu_0) \sigma_s(\bar{r}, \mu_0, E' \rightarrow E) d\mu_0 \quad . \quad (11)$$

To derive the discrete ordinates finite difference equation from Eq. 1, the directional flux and scattering cross section are replaced by the expansions, Eq. 2 and Eq. 10, on the right-hand side, i.e.,

$$\bar{\Omega} \cdot \nabla \psi(\bar{r}, \bar{\Omega}, E) + \Sigma_t(\bar{r}, E) \psi(\bar{r}, \bar{\Omega}, E)$$

$$= \frac{1}{4\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} \chi(E) v \sigma_f(\bar{r}, E') \left[\sum_{\ell=0}^\infty \sum_{n=-\ell}^\ell \psi_\ell^n(\bar{r}, E') Y_\ell^n(\bar{\Omega}') \right] d\theta' d\eta' dE'$$

$$+ \frac{1}{4\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} \sum_{p=0}^\infty (2p+1) \sigma_{s_p}(\bar{r}, E' \rightarrow E) P_p(\mu_0) \left[\sum_{\ell=0}^\infty \sum_{n=-\ell}^\ell \psi_\ell^n(\bar{r}, E) Y_\ell^n(\bar{\Omega}') \right] d\theta' d\eta' dE' .$$
(12)

Because of the orthogonality in Eq. 8a the first term on the right-hand side, if needed, can be rewritten:

$$\int_0^\infty \chi(E) v \sigma_f(\bar{r}, E') \int_{\bar{\Omega}'} Y_0^0(\bar{\Omega}') \sum_{\ell=0}^\infty \sum_{n=-\ell}^\ell Y_\ell^n(\bar{\Omega}') \psi_\ell^n(\bar{r}, E') d\bar{\Omega}' dE'$$

$$= \int_0^\infty \chi(E) v \sigma_f(\bar{r}, E') \psi_0^0(\bar{r}, E') dE' ,$$
(13)

where $\psi_0^0(\bar{r}, E')$ is the scalar flux.

To express the second term on the right-hand side in terms of η and θ instead of μ_0 , we note that the law of angular addition for Legendre polynomials is

$$P_\ell(\mu_0) = P_\ell(\eta') P_\ell(\eta) + 2 \sum_{n=1}^\ell \frac{(\ell-n)!}{(\ell+n)!} P_\ell^n(\eta') P_\ell^n(\eta) \cos n(\theta' - \theta) . \quad (14)$$

Substituting this result into the second term on the right-hand side of Eq. 12 we have

$$\begin{aligned}
& \frac{1}{4\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} \sum_{\ell=0}^\infty (2\ell+1) \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) P_\ell(\eta') P_\ell(\eta) \left[\sum_{\ell=0}^\infty \sum_{n=-\ell}^\ell \psi_\ell^n(\bar{r}, E') Y_\ell^n(\bar{\Omega}') \right] d\theta' d\eta' dE' \\
& + \frac{1}{4\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} \sum_{p=1}^\infty (2p+1) \sigma_{s_p}(\bar{r}, E' \rightarrow E) \left[2 \sum_{k=1}^p \frac{(p-k)!}{(p+k)!} P_p^k(\eta') P_p^k(\eta) \cos k(\theta' - \theta) \right] \\
& \times \sum_{\ell=0}^\infty \sum_{n=-\ell}^\ell \psi_\ell^n(\bar{r}, E') \left[\frac{(2\ell+1)(\ell-n)!}{(\ell+n)!} \right]^{\frac{1}{2}} P_\ell^n(\eta') \exp(in\theta') d\theta' d\eta' dE' \quad . \quad (15)
\end{aligned}$$

In rewriting the first term of Integral 15 we can use the fact that

$$(2\ell+1)^{\frac{1}{2}} P_\ell(\eta') = Y_\ell^0(\eta', \theta') \quad , \quad (16)$$

along with the orthogonality property Eq. 9 to write

$$\begin{aligned}
& \frac{1}{4\pi} \int_0^\infty \int_{-1}^1 \int_0^{2\pi} \sum_{\ell=0}^\infty (2\ell+1) \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) P_\ell(\eta') P_\ell(\eta) \sum_{\ell=0}^\infty \sum_{n=-\ell}^\ell \psi_\ell^n(\bar{r}, E') Y_\ell^n(\bar{\Omega}') d\theta' d\eta' dE' \\
& = \int_0^\infty \sum_{\ell=0}^\infty (2\ell+1)^{\frac{1}{2}} \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) \psi_\ell^0(\bar{r}, E') P_\ell(\eta) dE' \\
& = \int_0^\infty \sum_{\ell=0}^\infty \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) \psi_\ell^0(\bar{r}, E') Y_\ell^0(\bar{\Omega}') dE' \quad . \quad (17)
\end{aligned}$$

Note that in the second term of Integral 15 is equal to zero for $k \neq n$, since

$$\int_0^{2\pi} \cos k(\theta' - \theta) \exp(in\theta') d\theta' = 0 \quad , \quad k \neq n \quad . \quad (18)$$

For $k=n$ we have

$$\int_0^{2\pi} \cos k(\theta' - \theta) \exp(in\theta') d\theta' = \pi e^{in\theta} \quad (19)$$

Using this result when integrating the second term with respect to θ' we have

$$\begin{aligned} & \frac{1}{2} \int_0^\infty \int_{-1}^1 \sum_{n=1}^\infty \sum_{\ell=n}^\infty \sum_{p=n}^\infty (2p+1) \frac{(p-n)!}{(p+n)!} \sigma_{s_p}(\bar{r}, E' \rightarrow E) P_p^n(\eta') P_p^n(\eta) \\ & \times \left\{ \psi_\ell^n(\bar{r}, E') \left[\frac{(2\ell+1)(\ell-n)!}{(\ell+n)!} \right]^{1/2} P_\ell^n(\eta') e^{in\theta} \right\} d\eta' dE' \quad (20) \end{aligned}$$

Using the orthogonality condition

$$\int_{-1}^1 P_p^n(\eta') P_\ell^n(\eta') d\eta' = \begin{cases} \frac{2}{(2\ell+1)} \frac{(\ell+n)!}{(\ell-n)!} & p=\ell \\ 0 & \text{Otherwise} \end{cases} \quad (21)$$

for associated Legendre polynomials and integrating Eq. 20 with respect to η' we can write the second term as

$$\begin{aligned} & \int_0^\infty \sum_{\ell=1}^\infty \sum_{n=1}^\ell \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) \psi_\ell^n(\bar{r}, E') \left[\frac{(2\ell+1)(\ell-n)!}{(\ell+n)!} \right]^{1/2} P_\ell^n(\eta) e^{in\theta} dE' \\ & = \int_0^\infty \sum_{\ell=1}^\infty \sum_{n=1}^\ell \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) \psi_\ell^n(\bar{r}, E') Y_\ell^n(\eta, \theta) dE' \quad (22) \end{aligned}$$

Combining Eqs. 17 and 22 we can express the scattering integral as

$$\int_0^\infty \sum_{\ell=0}^\infty \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) \psi_\ell^0(\bar{r}, E') Y_\ell^0(\bar{\Omega}) + \sum_{\ell=1}^\infty \sum_{n=1}^\ell \sigma_{s_\ell}(\bar{r}, E' \rightarrow E) \psi_\ell^n(\bar{r}, E') Y_\ell^n(\bar{\Omega}) dE' \quad (23)$$

where

$$\psi_\ell^n(r, E') = \frac{1}{4\pi} \int_{-1}^1 \int_0^{2\pi} \psi(r, \bar{\Omega}, E') Y_\ell^n(\bar{\Omega}) d\theta d\eta \quad (24)$$

are the flux moments.

With the integrals on the right-hand side of Eq. 1 in the form of 13 and 14, the systems of dependant equations resulting from the discretization of the energy, direction and spatial variables may be solved within the nested iteration scheme in the DOT-IV code. The flux moments, therefore, are a consequence of the expansion of the scattering integral and are collected by DOT-IV and output according to a particular file structure format. The output file structure utilized by COUPL is the VARFLM file type in Fig. 6.

```

C*****VRFL
C          REVISED 10 NOV 76          -VRFL
C          -VRFL
CF          VARFLM          -VRFL
CE          VARIABLE MESH FLUX MOMENT DATA WITH BOUNDARY FLUXES          -VRFL
C          -VRFL
C*****VRFL
CD          ORDER OF GROUPS IS BY DECREASING ENERGY          VRFL
CD          I IS THE FIRST-DIMENSION INDEX          VRFL
CD          J IS THE SECOND-DIMENSION INDEX          VRFL
CD          JM=1 FOR 1-DIMENSIONAL GEOMETRY          VRFL
C-----VRFL
CS          FILE STRUCTURE          -VRFL
CS          -VRFL
CS          RECORD TYPE          PRESENT IF          -VRFL
CS          -----
CS          FILE IDENTIFICATION          ALWAYS          -VRFL
CS          FILE LABEL          ALWAYS          -VRFL
CS          FILE CONTROL          ALWAYS          -VRFL
CS          FILE INTEGER PARAMETERS          ALWAYS          -VRFL
CS          FILE REAL    PARAMETERS          ALWAYS          -VRFL
CS          -VRFL
CS          ***** (REPEAT OVER ALL GROUPS)          -VRFL
CS          *          FLUX MOMENTS          ALWAYS          -VRFL
CS          *          BOUNDARY DIRECTIONAL FLUX          ALWAYS          -VRFL
CS          *****          -VRFL
C          -VRFL
C-----VRFL
C-----VRFL
C-----VRFL
CR          FILE IDENTIFICATION          -VRFL
C          -VRFL
CL          HNAME, (HUSE(I), I=1,2), IVERS          -VRFL
C          -VRFL
CW          1+3*MULT=NUMBER OF WORDS          -VRFL
C          -VRFL
CD          HNAME          HOLLERITH FILE NAME - VARFLM - (A6)          -VRFL
CD          HUSE(I)          HOLLERITH USER IDENTIFICATION - (A6)          -VRFL
CD          IVERS          FILE VERSION NUMBER          -VRFL
CD          MULT          DOUBLE PRECISION PARAMETER          -VRFL
CD          1- A6 WORD IS SINGLE WORD          -VRFL
CD          2- A6 WORD IS DOUBLE PRECISION WORD          -VRFL
C          -VRFL
C-----VRFL
C          VRFL
C          VRFL

```

Fig. 6. VARFLM File Structure.

```

C-----VRFL
C                                         -VRFL
CR      FILE LABEL                               -VRFL
C                                         -VRFL
CL      DATE,USER,CHARGE,CASE,TIME,(TITL(I),I=1,12) -VRFL
C                                         -VRFL
CW      17*MULT=NUMBER OF WORDS                -VRFL
C                                         -VRFL
CD      DATE          AS PROVIDED BY TIMER OPTION 4 - (A6) -VRFL
CD      USER          AS PROVIDED BY TIMER OPTION 5 - (A6) -VRFL
CD      CHARGE         AS PROVIDED BY TIMER OPTION 6 - (A6) -VRFL
CD      CASE           AS PROVIDED BY TIMER OPTION 7 - (A6) -VRFL
CD      TIME           AS PROVIDED BY TIMER OPTION 8 - (A6) -VRFL
CD      TITL(I)        TITLE PROVIDED BY USER      - (A6) -VRFL
C                                         -VRFL
C-----VRFL
C                                         VRFL
C                                         VRFL
C-----VRFL
CR      FILE CONTROL                               -VRFL
C                                         -VRFL
CD      IGM,NEUT,JM,LM,IMA,MMA,ISM,IMSISM,ISBT,ITER,(IDUM(N),N=1,15) -VRFL
C                                         -VRFL
CW      25=NUMBER OF WORDS                      -VRFL
C                                         -VRFL
CD      IGM            NUMBER OF ENERGY GROUPS      -VRFL
CD      NEUT           LAST NEUTRON GROUP             -VRFL
CD                        (IGM IF ALL NEUTRONS, 0 IF ALL GAMMAS) -VRFL
CD      JM            NUMBER OF SECOND-DIMENSION (J) INTERVALS -VRFL
CD      LM            MAXIMUM LENGTH OF MOMENT EXPANSION -VRFL
CD      IMA           MAXIMUM NUMBER OF FIRST-DIMENSION INTERVALS -VRFL
CD      MMA           NUMBER OF BOUNDARY DIRECTIONS   -VRFL
CD      ISM           NUMBER OF I-BOUNDARY SETS       -VRFL
CD      IMSISM        TOTAL NUMBER OF I-INTERVALS, ALL I-SETS -VRFL
CD      ISBT          I-SET FOR SYSTEM BOUNDARIES    -VRFL
CD      ITER          OUTER ITERATION NUMBER AT WHICH FLUX WAS -VRFL
CD                        WRITTEN                     -VRFL
CD      IDUM(I)       ARRAY SET TO 0                 -VRFL
C                                         -VRFL
C-----VRFL
C                                         VRFL
C                                         VRFL
C-----VRFL
CR      FILE INTEGER PARAMETERS                   -VRFL
C                                         -VRFL
CL      (LMBIG(IG),IG=1,IGM),                     -VRFL
CL      *(IMBIS(IS),IS=1,ISM),(ISET(J),J=1,JM)     -VRFL
C                                         -VRFL
CW      IGM+ISM+JM=NUMBER OF WORDS                 -VRFL
C                                         -VRFL
CD      LMBIG(IG)     LENGTH OF MOMENT EXPANSION FOR GROUP IG -VRFL
CD      IMBIS(IS)     NUMBER OF INTERVALS IN ISET IS  -VRFL
CD      ISET(J)       I-SET ASSIGNED TO INTERVAL J   -VRFL
C                                         -VRFL
C-----VRFL
C                                         VRFL
C                                         VRFL

```

Fig. 6. (continued).

```

C-----VRFL
CR      FILE REAL PARAMETERS                      -VRFL
C                                             -VRFL
CL      (Z(J),J=1,JM1),((R(I,IS),I=1,IM1),IS=1,ISM), -VRFL
CL      *(ENER(IG),IG=1,IGM),EMIN,ENEUT,EV,DEVDK,EFFK,POWER, -VRFL
CL      *(DUMRL(I),I=1,13)                      -VRFL
C                                             -VRFL
CW      JM+IMSISM+ISM+IGM+20=NUMBER OF WORDS      -VRFL
C                                             -VRFL
CD      Z(J)                J-INTERVAL BOUNDARIES -VRFL
CD      R(I,IS)              I-INTERVAL BOUNDARIES FOR I-SET I -VRFL
CD      ENER(IG)              TOP ENERGY BOUNDARY OF GROUP IG -VRFL
CD      EMIN                  BOTTOM ENERGY BOUNDARY OF GROUP IGM -VRFL
CD      ENEUT                  BOTTOM ENERGY BOUNDARY OF GROUP NEUT -VRFL
CD                        (0 IF NEUT=0)            -VRFL
CD      EV                    EIGENVALUE           -VRFL
CD      DEVDK                  DERIVATIVE OF EV VS. EFFK -VRFL
CD      EFFK                    EFFECTIVE MULTIPLICATION FACTOR -VRFL
CD      POWER                  POWER (WATTS) TO WHICH FLUX IS NORMALIZED -VRFL
CD      DUMRL                  ARRAY SET TO 0        -VRFL
CD      JM1                    JM+1                 -VRFL
CD      IM1                    IMBIS(IS)+1          -VRFL
C                                             -VRFL
C-----VRFL
                                             VRFL
                                             VRFL
                                             VRFL
                                             VRFL
C-----VRFL
CR      FLUX MOMENTS                      -VRFL
C                                             -VRFL
CL      ((FLUM(I,L),I=1,IMS),L=1,LMS)        -VRFL
C                                             -VRFL
CW      IMS*LMS=NUMBER OF WORDS              -VRFL
C                                             -VRFL
C      DO 1 J=1,JM                          -VRFL
C      1 READ(N)    *LIST AS ABOVE*          -VRFL
C                                             -VRFL
CD      FLUM                FLUX BY INTERVAL AND MOMENT INDEX -VRFL
CD      IMS                  IMBIS(IS) FOR IS CORRESPONDING TO J -VRFL
CD      LMS                  LMBIG(IG)         -VRFL
C                                             -VRFL
C-----VRFL
                                             VRFL
                                             VRFL

```

Fig. 6. (continued).

C-----			VRFL
CR	BOUNDARY DIRECTIONAL FLUX		-VRFL
C			-VRFL
CL	((FIO(M,J),M=1,MMA),J=1,JM), ((FJO(M,I),M=1,MMA),I=1,IMB)		-VRFL
C			-VRFL
CW	MMA*(JM+IMB)=NUMBER OF WORDS		-VRFL
C			-VRFL
CD	FIO	DIRECTIONAL FLUX OUTGOING BY DIRECTION AND	-VRFL
CD		J-INTERVAL	-VRFL
CD	FJO	DIRECTIONAL FLUX OUTGOING BY DIRECTION AND	-VRFL
CD		I-INTERVAL. FJO=0 FOR A 1-D GEOMETRY	-VRFL
CD	IMB	IMBIS(IS) FOR IS CORRESPONDING TO ISBT	-VRFL
C			-VRFL
C-----			VRFL
			VRFL
			VRFL
END			VRFL
			VRFL

Fig. 6. (continued).

III. STATEMENT OF THESIS PROBLEM

The objective for this thesis is to create and document a coupling code to extend the dimensional limitations of the DOT-IV two-dimensional discrete ordinates transport code to the case of intersecting perpendicular cylinders. A general problem of this type is shown in Fig. 7.

A neutron source is set on the disc surface, A, in Fig. 7. A DOT-IV calculation is run along the vertical cylinder. The code, COUPL, must couple the results of the leg 1 calculation through an interface, B, to serve as a boundary source for a DOT-IV calculation in leg 2. The unknown fluxes in leg 2 can then be computed via a second DOT-IV calculation.

The accuracy of the code is evaluated by comparing scalar fluxes $\psi_0(\bar{r}, E)$, where

$$\psi_0(\bar{r}, E) = \int_{\bar{\Omega}} \psi(\bar{r}, \bar{\Omega}, E) d\bar{\Omega} \quad , \quad (25)$$

at points along the leg 2 centerline with corresponding results from a MORSE Monte Carlo three-dimensional calculation.

ORNL-DWG 81-17716

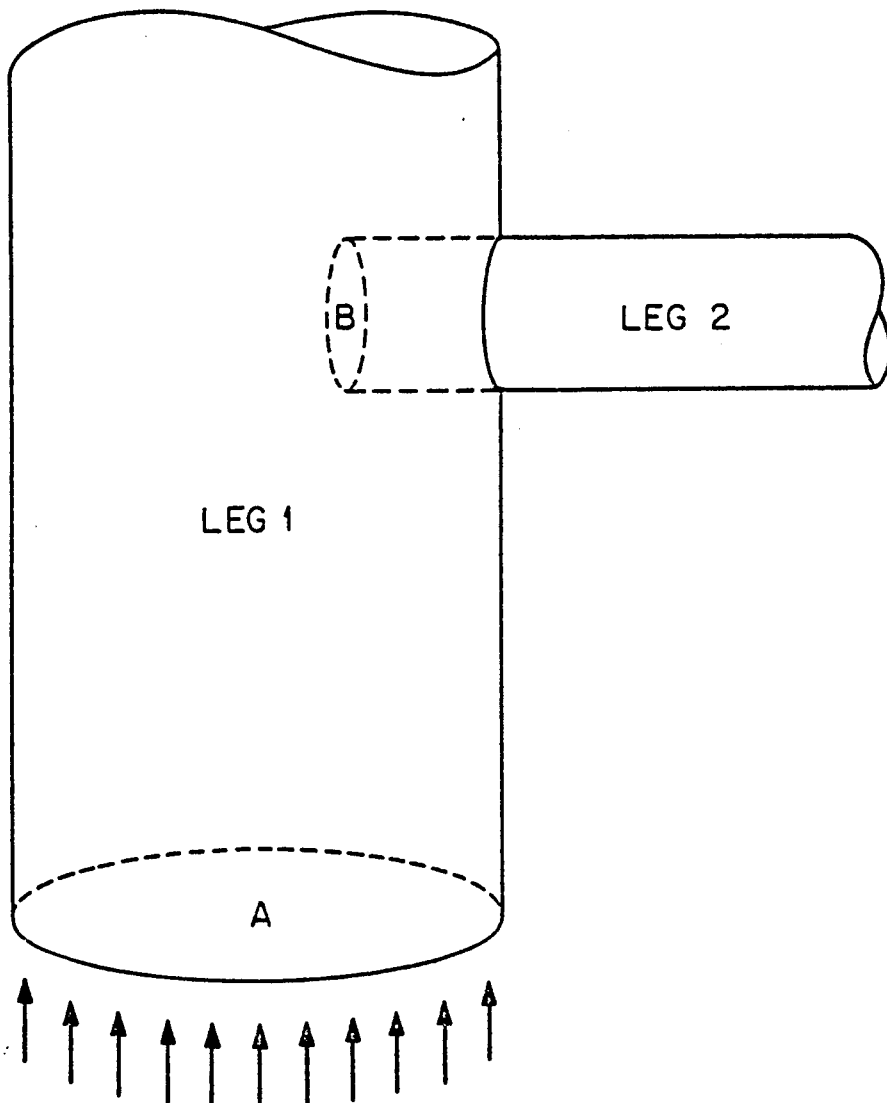


Fig. 7. Intersecting Perpendicular Cylinders.

IV. COUPL--COUPLING DOT-IV CALCULATIONS THROUGH A 90- DEGREE ROTATION OF THE r - z COORDINATE SYSTEM

COUPL is a fortran program designed to transform the VARFLM DOT-IV output file to a BNDRYS DOT-IV input file in a region of interest. This affectively couples the end of a second DOT-IV cylinder (leg 2) into the side of the first, already solved, DOT-IV cylinder (leg 1) as in Fig. 7.

The general coupling situation for cylindrical DOT-IV problems is shown in Fig. 8. The r - θ disc surface on the intersecting end of leg 2 is the surface on which a boundary source is required in order to start a leg 2 DOT-IV calculation. COUPL constructs this boundary source based on the known leg 1 solutions in the interface region. The user may choose any spatial mesh and quadrature set for the leg 2 system. In addition, the location of the leg 2 boundary source surface may be chosen by the user such that the distance from the leg 1 centerline to the center of the leg 2 source surface along the leg 2 centerline varies from zero to some positive value which will not cause certain peripheral points on the leg 2 boundary source surface to lie 'outside' the leg 1 system. In this way the problem may be run as a simple 'bootstrapping' problem where the leg 2 disk source surface lies tangent to the outer cylindrical surface of leg 1 as in Fig. 9 or, more generally, as an imbedded mesh problem, Fig. 8, with the leg 2 source surface at least partially contained on the leg 1 system. The number of leg 2 boundary source points which lie 'outside' the leg 1 system will be a function of the offset distance, RDC in Fig. 8, and the diameter of leg 2 relative to that of leg 1. For reasons of flexibility, some portions of the leg 2 boundary source surface are allowed to lie outside the leg 1 boundaries as long as they

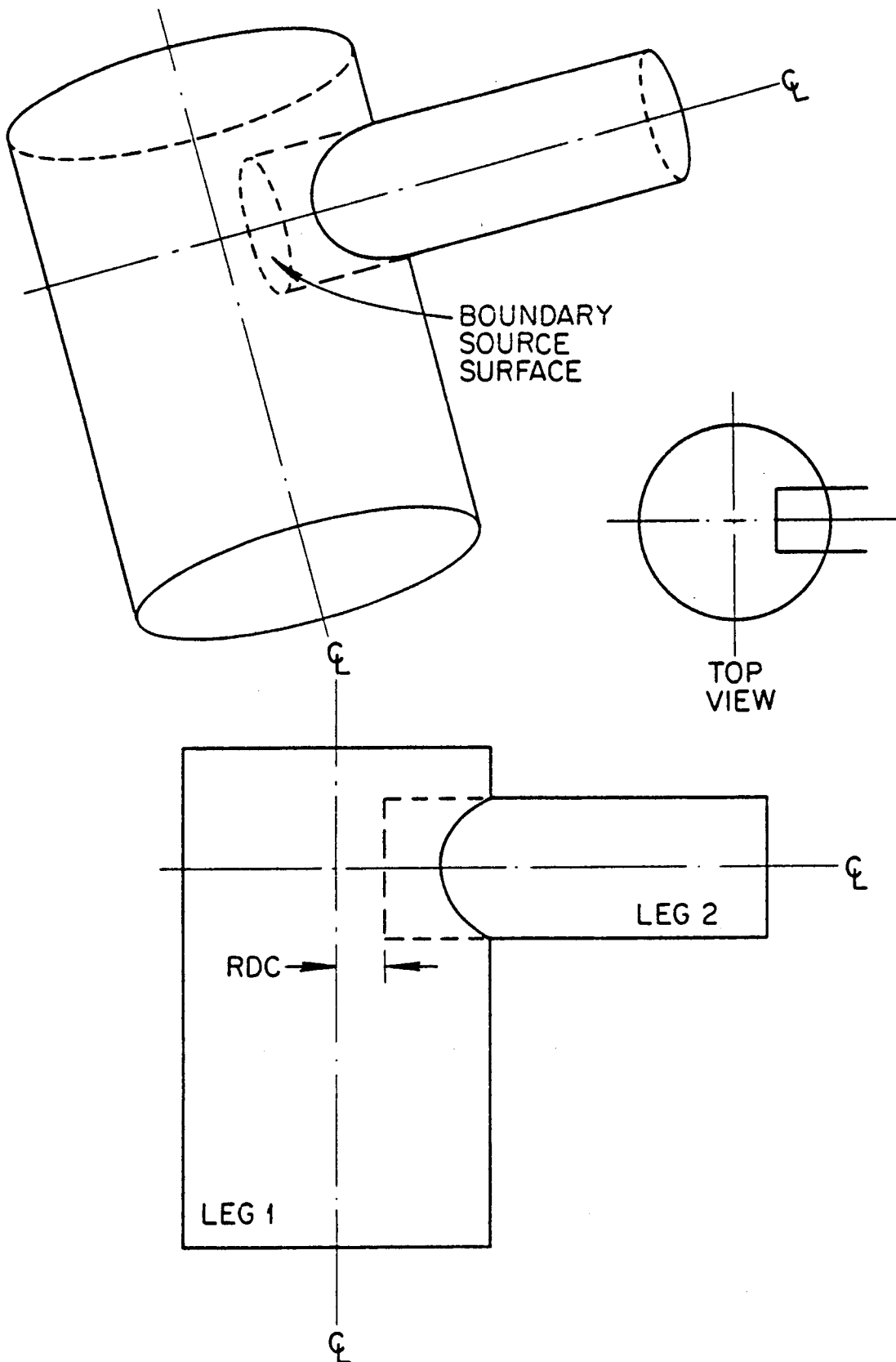


Fig. 8. Mesh Embedded Boundary Source.

ORNL-DWG 81-17717AR

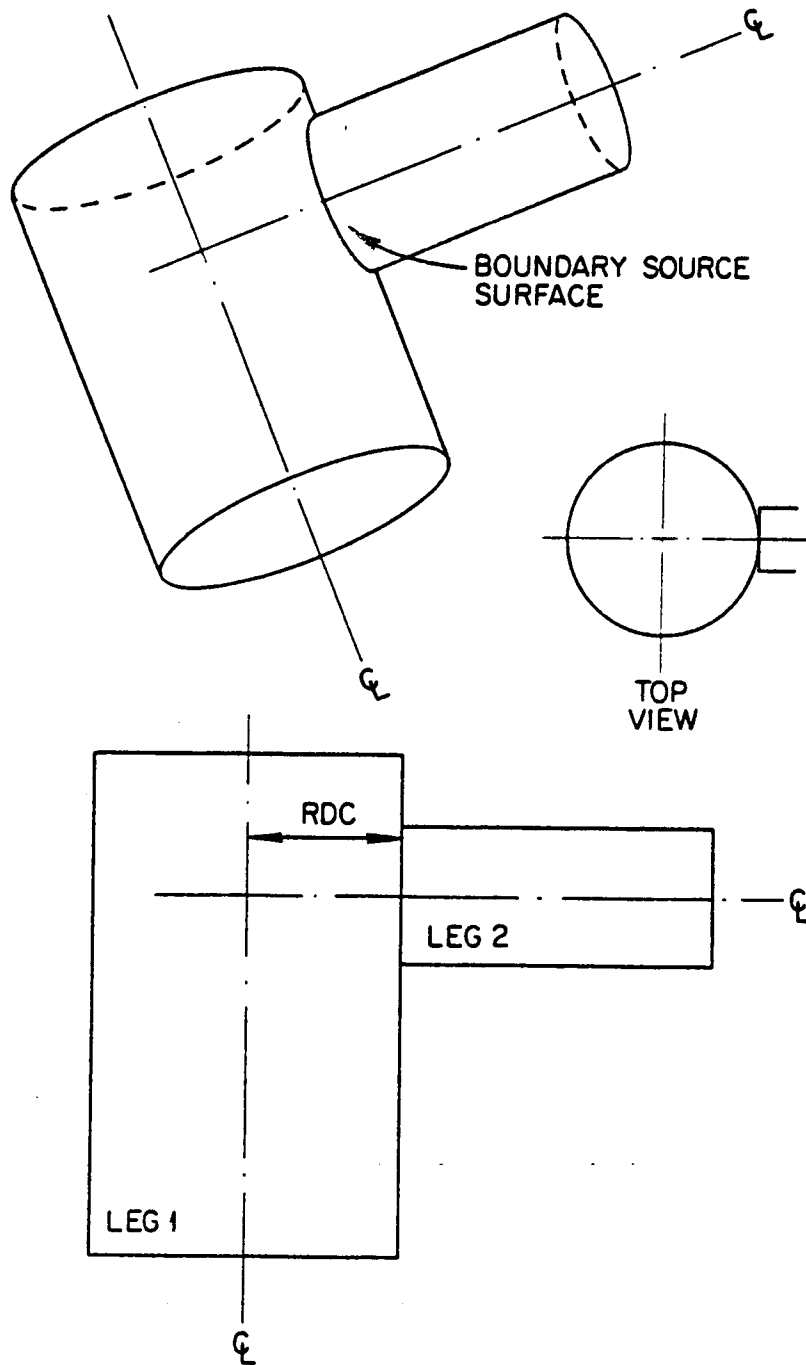


Fig. 9. 'Bootstrapping' Technique.

are within a tolerance distance, TOL (a fraction of a spatial interval which may be user specified), of the leg 1 boundaries. Obviously, the accuracy of the coupling process may be reduced by allowing TOL to be too large.

The method used by COUPL to construct the leg 2 boundary source involves forming a table whereby each mesh point on the leg 2 source surface is mapped into the corresponding leg 1 mesh interval which contains that point. In the case where certain leg 2 points are on the surface of or on the outside of the leg 1 system, it suffices to pick the closest leg 1 interval to the leg 2 point. Points on the leg 2 source surface are in $r-\theta$ (resembling a pie) geometry while the leg 1 intervals are in $r-z$ geometry (resembling stacked sets of nested rings).

We shall refer to the leg 2 system as the 'primed' system, (r',z') , where $z'=c=\text{const.}$ on the source surface and θ is the azimuthal angle (measured from verticle as shown in Fig.10) and leg 1 as the unprimed system (r,z) (because DOT-IV is a two-dimensional code, i.e., the solution consists of sets of flux moments which depend only on spatial interval and energy we need not consider an azimuthal variable in the leg 1 system).

Since the (r,z) coordinates of the $r'=0$ center point (origin) in the leg 2 source surface are provided (RDC and H2), we can easily find exact leg 1 coordinates, RE and ZE, of any point on the leg 2 boundary surface using the formula

$$\begin{aligned} RE &= \sqrt{RDC^2 + (r' \sin \theta)^2} \\ ZE &= H2 + r' \cos \theta \end{aligned} \quad . \quad (26)$$

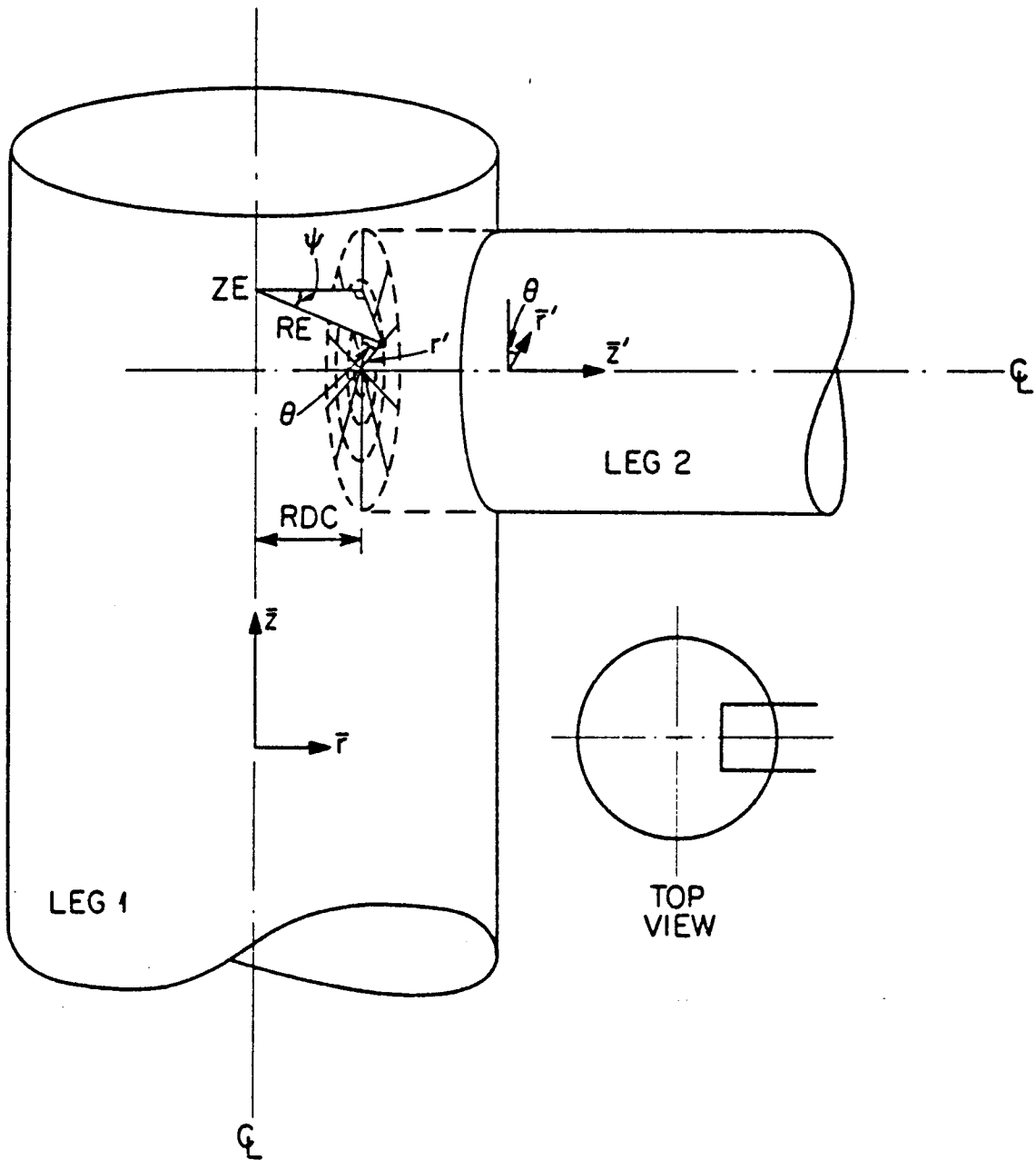


Fig. 10. Transforming Position.

Once the exact leg 1 coordinates are known for each point on the leg 2 source surface we must know which leg 1 spatial interval contains each of the transformed points (DOT-IV solutions in leg 1 being given by interval). This scanning procedure is done in subroutine LOCAT. Here, the table is constructed in which, given the indices of a leg 2 source surface mesh point, the corresponding flux moment solution set is stored. Another table array which is calculated in this step is the angle ψ . It is the angle between the radius, RE , directed to each leg 2 source surface mesh point, and the leg 2 centerline in Fig.10. This angle is necessary in the following transformation of direction because the directional cosine μ , Fig. 3, in the direction set for r-z geometry is made with respect to the r axis which points to the corresponding source surface point from the leg 1 centerline.

The next step in constructing a leg 2 boundary source is to transform each leg 2 direction (at each boundary surface point) into leg 1 coordinates. The spherical harmonics are then evaluated at these angles and expanded with the flux moment set which corresponds to that point (Eq. 2). This is a convenient process in that there is no need to search the leg 1 quadrature to find the nearest angle to the exact transformation of the leg 2 directional set as was the case with the spatial transformation. The flux moments are a discrete function of position while the spherical harmonics are a continuous function of direction.

The directional transformation is straightforward. In the primed leg 2 system the direction pair (μ', η') is used to find the third direction $\pm\xi'$ (Fig. 3). Although flux is symmetrical about the third directional axis, θ , $(\mu', \eta', +\xi')$ and $(\mu', \eta', -\xi')$ will in general be

transformed to a directional pair in the leg 1 system with differing μ and θ . The pair of fluxes corresponding to this directional pair are averaged into one value which is assigned to each of the original leg 2 boundary source directions $(\mu', \eta', +\xi')$ and $(\mu', \eta', -\xi')$ to preserve symmetry.

The transformation of each direction may be considered as an orthogonal rotational transformation of the unit vector $\bar{\Omega}'$. (In Fig. 11 x-y-z axes are used for generality and the variables corresponding to the geometry may be substituted before and after the transformation.)

A matrix T is sought which rotates the primed axes through the angle θ about the y' axis and through the angle ψ about the z'' axis. These may be thought of as separate rotations T_1 and T_2 , respectively. T_1 rotates the primed axis system to match an intermediate double primed system, x''-y''-z'', in Fig. 11. T_2 rotates the intermediate double primed system through the angle ψ about the z'' axis.

Letting $\bar{\Omega}'$ be the unit direction vector in the primed system whose components, x'-y'-z', have been substituted appropriately for the corresponding quadrature components (ξ', η', μ') , we have

$$\text{for } \bar{\Omega}' = \begin{bmatrix} z' \\ y' \\ x' \end{bmatrix}, \text{ where } \begin{cases} \mu' = z' \\ \eta' = y' \\ \xi' = x' \end{cases},$$

$$\bar{\Omega}'' = T_1 \bar{\Omega}'$$

$$\begin{bmatrix} z'' \\ y'' \\ x'' \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} \quad (27)$$

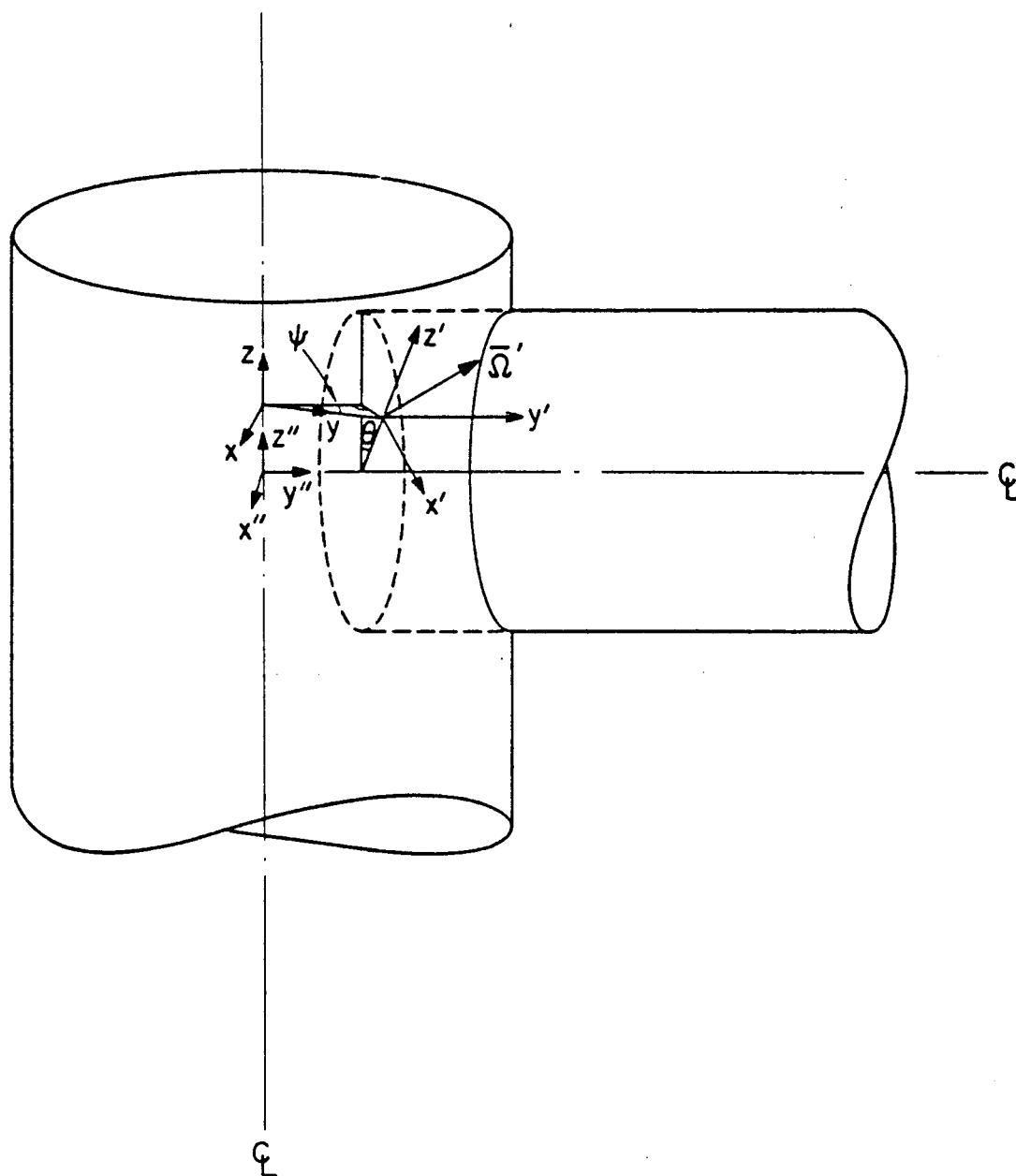


Fig. 11. Transforming Direction.

and

$$\bar{\Omega} = T_2 \bar{\Omega}'$$

$$\begin{bmatrix} z \\ y \\ x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} z'' \\ y'' \\ x'' \end{bmatrix} \quad (28)$$

From this we see that

$$\begin{aligned} \begin{bmatrix} z \\ y \\ x \end{bmatrix} &= T_1 T_2 \begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} = T \begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} \\ \begin{bmatrix} z \\ y \\ x \end{bmatrix} &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & \sin \theta \sin \psi & -\sin \theta \cos \psi \\ 0 & \cos \psi & \sin \psi \\ \sin \theta & -\cos \theta \sin \psi & \cos \psi \cos \psi \end{bmatrix} \begin{bmatrix} z' \\ y' \\ x' \end{bmatrix} . \end{aligned}$$

$$\text{In leg 1} \begin{cases} \eta = z \\ \mu = y \\ \xi = x \end{cases} . \quad (29)$$

In substituting the appropriate directional component notation in the unprimed leg 1 system, note that the axial component is now z which corresponds to η and the radial component is now y which corresponds to μ . This permutation is due to 90-degree rotation about both the y and the z axes which would have been necessary to consider in constructing T had we oriented the x - y - z axes with respect to the leg 1 cylinder in the same manner as the x' - y' - z' axes were oriented with respect to the leg 2 cylinder. If we preserve the orientation of the axes with respect to their corresponding cylinders we can write

$$\begin{aligned}
T_P &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} T \\
&= \begin{bmatrix} 0 & \cos\psi & \sin\psi \\ \cos\theta & \sin\theta \sin\psi & -\sin\theta \cos\psi \\ \sin\theta & -\cos\theta \sin\psi & \cos\theta \cos\psi \end{bmatrix}, \\
\text{and } \begin{bmatrix} \mu \\ \eta \\ \xi \end{bmatrix} &= T_P \begin{bmatrix} \mu' \\ \eta' \\ \xi' \end{bmatrix}. \tag{30}
\end{aligned}$$

The complete transformation is therefore composed of four rotations.

After a leg 1 direction has been found for a given point on the leg 2 boundary source surface, the directional flux at that point and direction may be approximated by the expansion in Eq. 2.

COUPL scans the r' - θ boundary surface first by azimuthal angle from 0 to π , then by leg 2 direction, and then by radius level from 0 to R_2 (radius of leg 2) and finally by energy level. At every r' level the computed directional fluxes for each leg 2 direction are accumulated and averaged over azimuthal angle, θ . Here it is only necessary to average over azimuthal angles ranging from 0 to π due to the leg 1 azimuthal symmetry in the VARFLM solution set. Each averaging produces a directional flux for that r' level for every energy group until the source is complete. Finally, COUPL writes a DOT-IV boundary source in BND RYS format (Fig. 5). A brief discussion of the access to COUPL is given in Appendix C.

V. NUMERICAL TESTING: RESULTS AND COMPARISONS

To test the accuracy of COUPL a typical test problem (Fig. 12) was solved in which the two intersecting pipes, leg 1 and leg 2, were of equal diameter. The pipes were modeled as having a 5 cm. concrete shell thickness with a 50 cm. outer radius. Leg 1 ends with a 5-cm-thick concrete cap. Otherwise, the two-legged duct is open ended, filled with sodium and surrounded by void.

A fourteen energy group isotropic boundary source was placed at the bottom end ($z=0$) of leg 1, and a DOT-IV calculation performed in the first leg. This created a solution set for leg 1 in the VARFLM format, i.e., consisting of flux moment sets at each r - z spatial interval. A COUPL calculation was performed on the leg 2 boundary source interface creating a boundary source there. This enabled a final DOT-IV calculation down leg 2 of the duct system and solutions in VARFLM format for each r' - z' interval.

To visually clarify the results, level curves were plotted for the DOT-IV scalar fluxes in both legs. The level curves (isoplots) are shown below (Figs. 13 and 14).

One effect shown in the level curve plots is the so-called 'ray effect'. This is characterized by peaks and valleys in the level curves caused by truncation error in the moment expansion, (Eq. 2), which is higher in some directions than in others. This effect has been reduced somewhat in these plots by using a finer (larger) quadrature set (96 directions) for all calculations.

A Monte Carlo code, MORSE,³ was run throughout the entire two-legged duct system using the same leg 1 boundary source as the first DOT-IV calculation. MORSE, a widely used (though expensive) three-dimensional code,

ORNL-DWG 81-17721

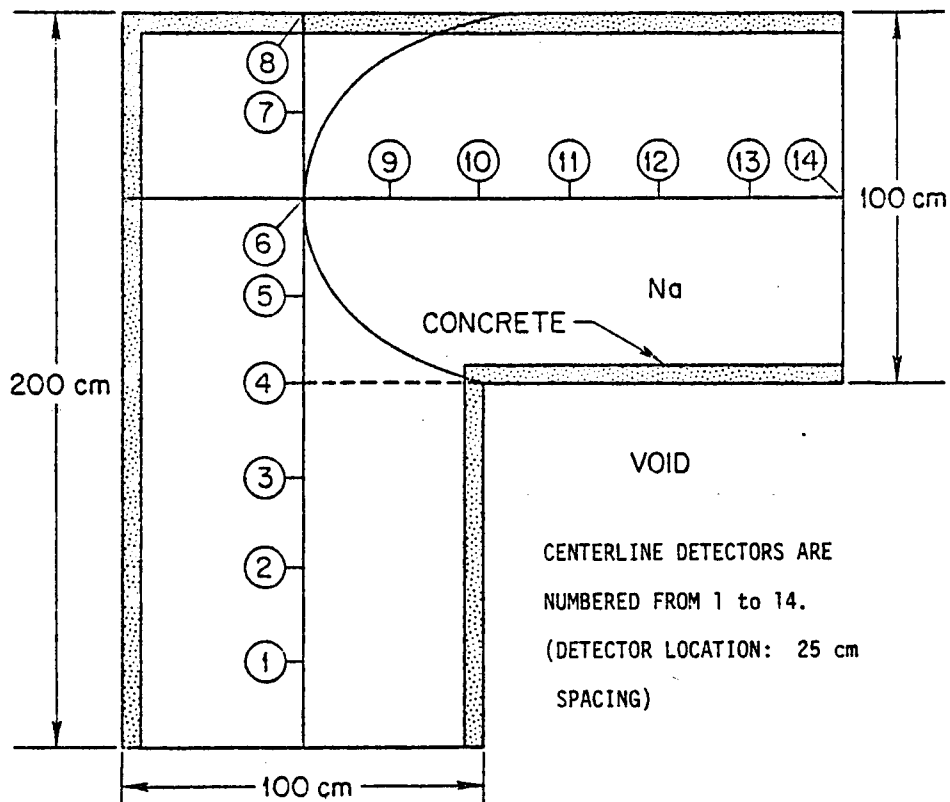


Fig. 12. COUPL Test Problem.

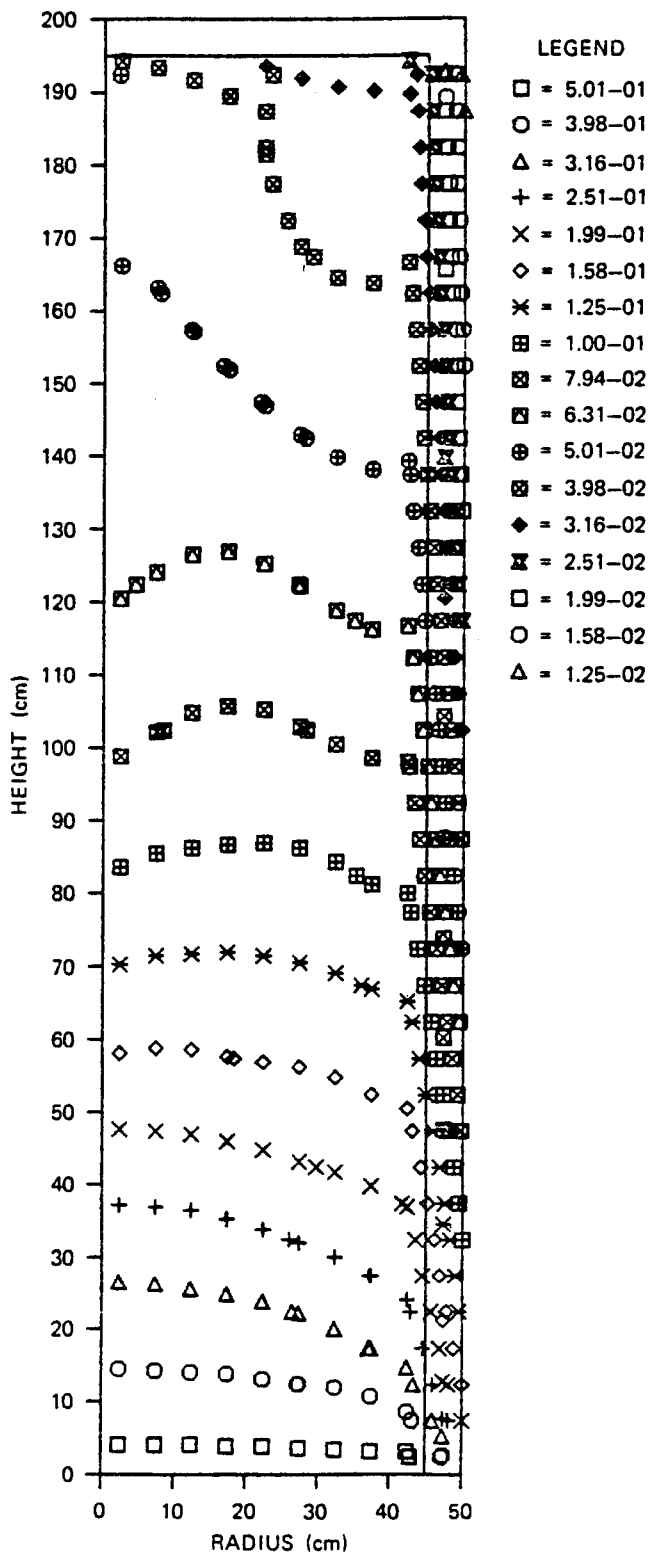


Fig. 13. Isoplot--Leg 1.

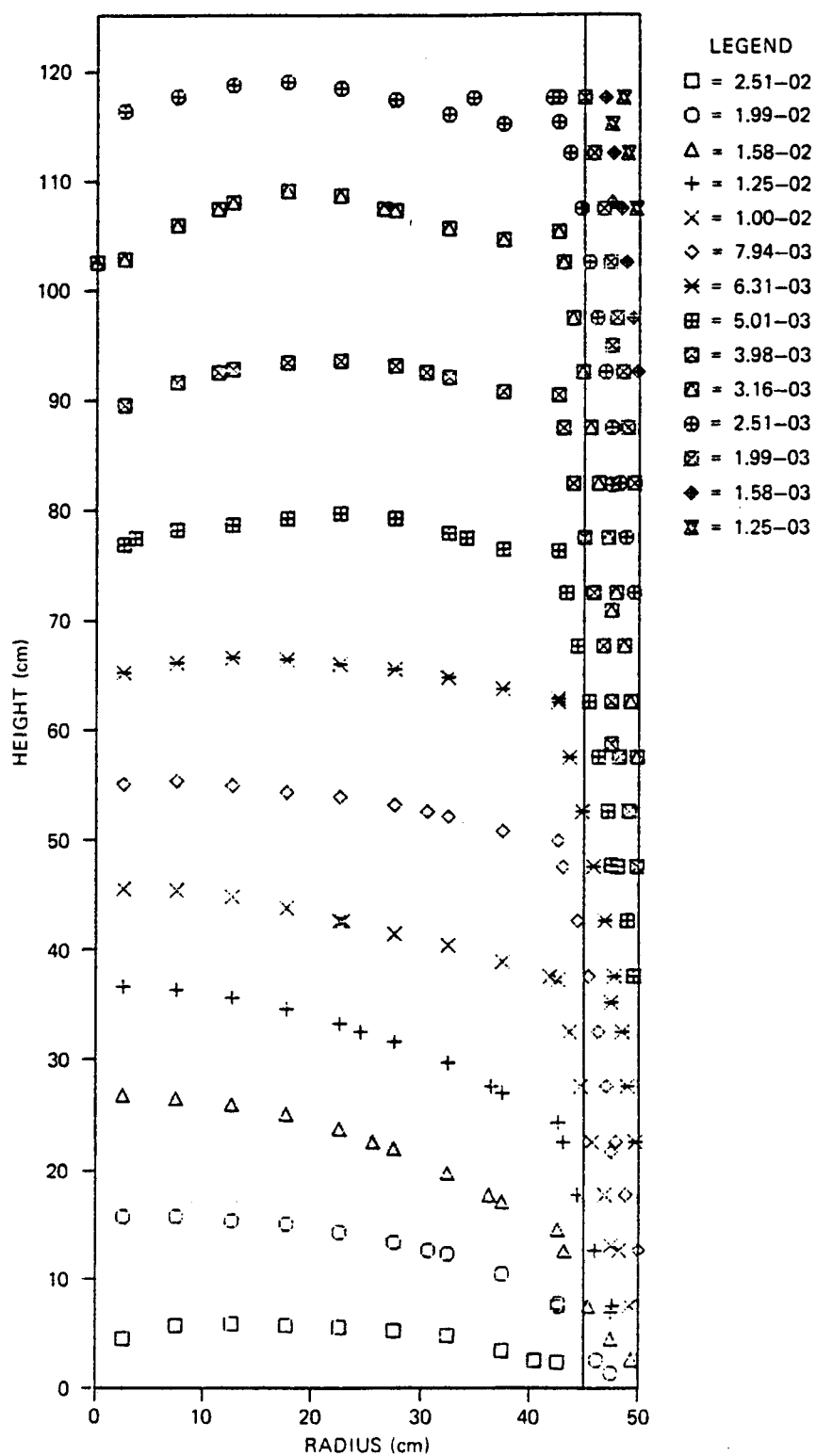


Fig. 14. Isoplot--Leg 2.

seemed to be a reasonable means to evaluate the validity of the results from the coupled DOT-IV runs. To hold down operating costs the same fourteen group isotropic source was used, and the number of intervals in the spatial mesh was fairly small; ten radial and forty axial intervals in leg 1, and ten radial and thirty axial intervals in leg 2. A total of 14 flux-response detectors were placed along the centerline of each leg in the two-legged system at the locations denoted in Fig. 12, p. 41. The scalar flux,

$$\psi_0(\bar{r}, E) = \int_{\bar{\Omega}} \psi(\bar{r}, \bar{\Omega}, E) d\bar{\Omega} ,$$

from each of the detectors in the MORSE calculation was summed over all fourteen groups and compared with the scalar flux from the corresponding DOT-IV intervals. The precision of the MORSE results at each detector is represented as the total fractional standard deviation for each detector response. These numbers, output by the code as an indicator of the precision, must be small (approximately 0.1% or less) before reasonable accuracy can be assumed. From the detector response values, flux attenuation curves were plotted for both legs using each method.

Table 2 below contains the total(summed over group)scalar flux values calculated at the detector locations using each method along with the percent error between the coupled DOT-IV solutions and the MORSE solutions at each detector. The MORSE total fractional standard deviations are also given for each detector.

The attenuation curves (Fig. 15) were drawn from the data in Table 2 with RDC=25 cm. Note the increasing separation of the curves in leg 1 as they near the 200 cm mark. This is due mainly to modeling differences. MORSE, being three-dimensional, allows for the modeling of the leg 2

Table 2. Test Results at Centerline Detectors
(Leg 1 and 2); Total Scalar Flux.

DETECTOR NO.	MORSE	DOT-IV	MORSE F.S.D.	% ERROR
1	3.4505255E-01	3.4045E-01	0.01162	1.333
2	2.0393890E-01	1.9984E-01	0.01300	2.01
3	1.2804044E-01	1.2002E-01	0.01481	6.26
4	8.5740090E-02	8.0784E-02	0.01768	5.78
5	6.0925605E-02	6.2107E-02	0.02141	1.94
6	4.5672865E-02	5.3816E-02	0.02539	17.82
7	3.7417250E-02	4.9280E-02	0.03671	31.70
8	1.9277328E-02	4.2277E-02	0.46469	119.31
9 [‡]	4.3028845E-02	leg2 source(RDC=25cm)		
10	3.4848660E-02	1.7354236E-02	0.02717	50.20
11	1.3982142E-02	9.5097976E-03	0.03417	31.99
12	5.7983835E-03	5.4471952E-03	0.04991	6.06
13	3.6746320E-03	3.4533E-03	0.06527	6.02
14	2.2940645E-03	2.3680E-03	0.07680	3.22

* Fractional Standard Deviation.

† This detector lies on the leg 2 boundary source (created by COUPL) when RDC=25 cm.

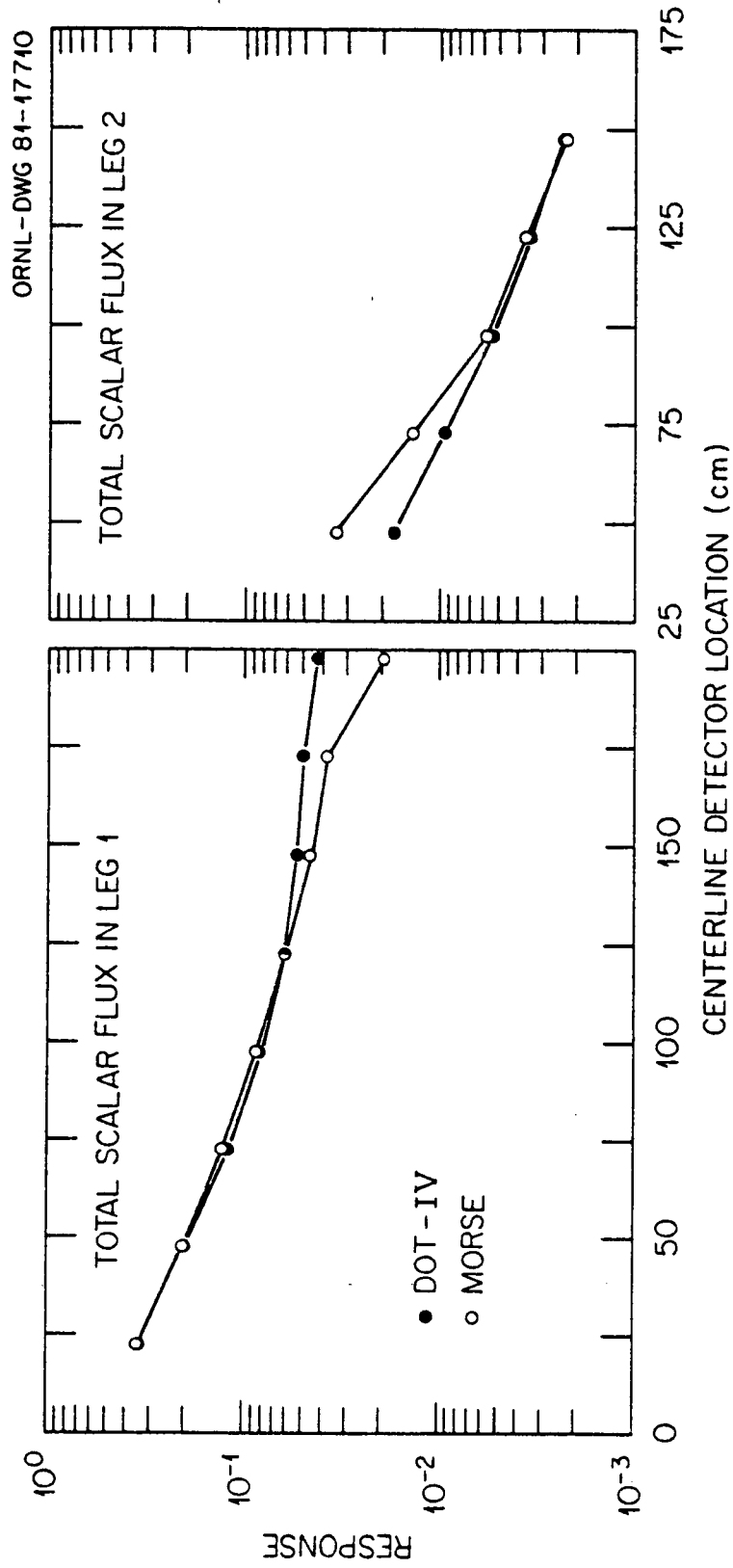


Fig. 15. Scalar Flux Attenuation Curves With
RDC=25 cm: Leg 1 and Leg 2.

opening in the top end of leg 1 shell. A considerable number of particles pass through this opening into leg 2. DOT-IV, on the other hand, can model only in two dimensions and must model the top of the leg 1 cylinder as having a uniform (constant) shell material with respect to azimuthal rotations about its' centerline. For this reason, particles which should be escaping into leg 2 are being reflected (scattered) back into leg 1 by the concrete shell which is present where the leg 2 opening should be. Therefore, the MORSE responses drop off more rapidly at the end of leg 1 than do the DOT-IV responses.

In the leg 2 curves, however, note that the situation has reversed. The MORSE responses are higher than those of DOT-IV. The reason for this is again attributed to modeling differences (Fig.16). Because the source surface at the end of the leg 2 cylinder can be created at various locations (depending upon the value of RDC) along the leg 2 centerline which are embedded in the leg 1 spatial mesh, there may be a certain amount of leg 2 radial boundary surface inside of leg 1 which, influenced by flux there, contributes to the response at the detectors. Particles passing through this surface in the MORSE calculation would not be considered in the coupled DOT-IV calculation. Furthermore, in the leg 2 calculation, DOT-IV utilizes an azimuthally constant concrete shell which is modeled for the entire length of leg 2. In MORSE there is no such overlap of geometries.

This response discrepancy was noted to vary with changes in RDC. When $RDC=0$ and more leg 2 radial boundary surface is embedded in leg 1, the difference in response values at detector #10 grew larger. When $RDC=50$ cm, none of the leg 2 radial boundary surface would be imbedded

ORNL-DWG 81-17722

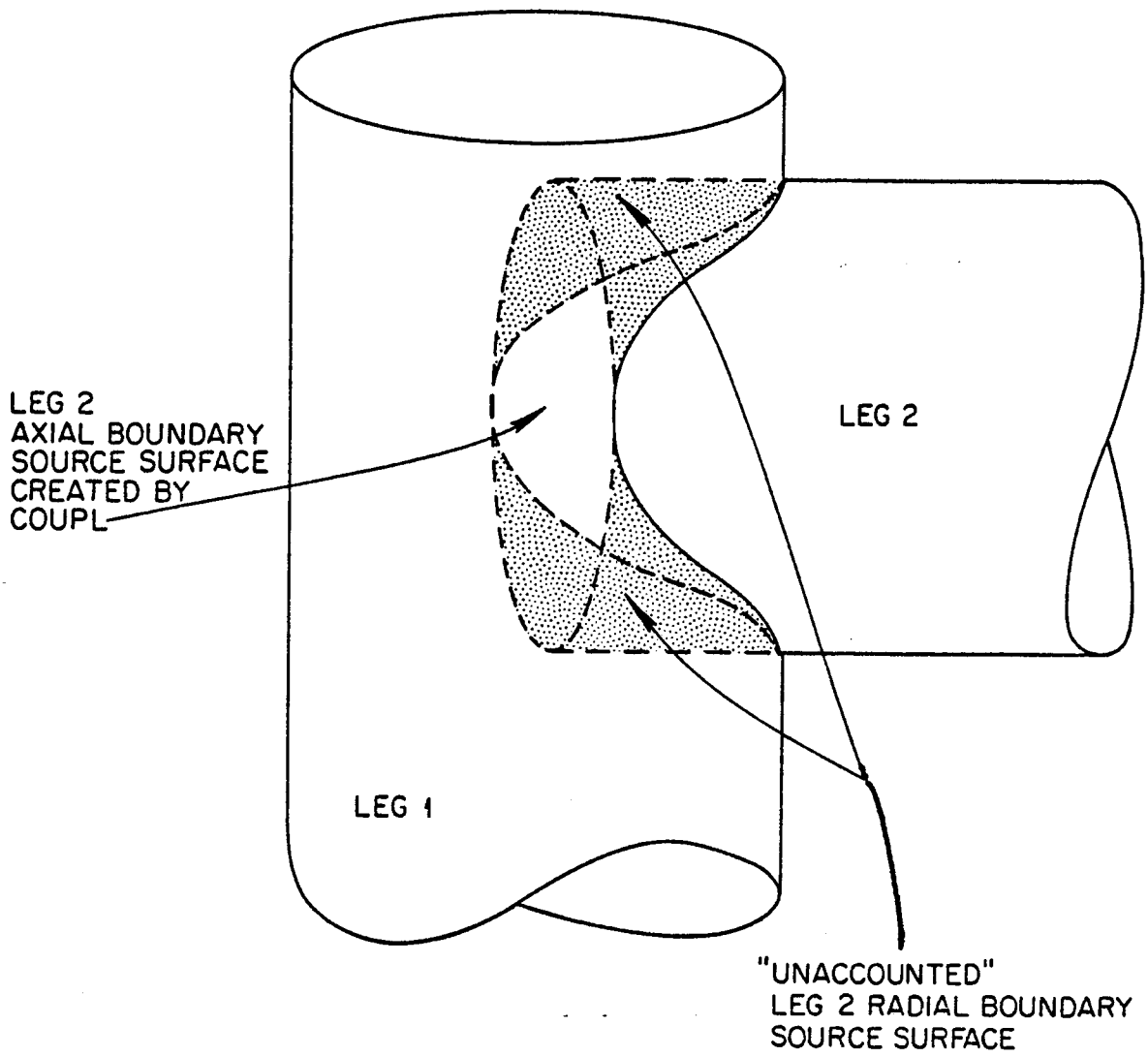


Fig. 16. A Modeling Problem.

in leg 1 but values chosen to construct the leg 2 source surface would come from the concrete shell region of leg 1 at the interface of leg 2. This concrete does not exist in the MORSE calculation and provides source values which are too small. This can be seen in the level curves of Fig. 13, p. 42.

The proper selection of RDC involves placing the boundary source for leg 2 neither too 'near' the concrete shell of leg 1 nor too 'near' the leg 1 centerline, and thus exposing more of the leg 2 radial boundary surface to leg 1 flux. RDC=25 cm seemed to produce reasonable results.

Although the 'worst' percent error with respect to the MORSE calculation was approximately 50%, the average percent error over all centerline detectors was approximately 11.9% for RDC=25 cm. The shape of the coupled DOT-IV attenuation curves seem to resemble that of the MORSE attenuation curves: they seem to converge as they move down the centerline. For all detectors except those closest to the interface region, the percent error is reasonably low, approximately 6%.

What is most important, the agreement between the coupled calculation and the MORSE calculation improves toward the end of leg 2. Specifically, the percent error at the end of leg 2 (detector 14, Table 2, p. 45) is approximately 3%.

The central processing time required by COUPL for a job of the size of the test problem is approximately 1 min, 26 sec, with 2.7 I/O's. Storage is 336K bytes (at most) for any one job step. The total job cost for the two DOT-IV calculations and the COUPL calculation is approximately 53% of the job cost of the MORSE calculation.

VI. CONCLUSIONS

If the primary concern of the user is flux values in leg 2 which are 'near' the interface region (e.g., detectors 9, 10, or 11 in Fig. 12, p. 41), then a MORSE calculation is recommended. If, however, the user wishes to estimate flux levels near detectors 12, 13, or, especially to estimate the number of particles which cross the boundary at the end of leg 2 (near detector 14 in Fig. 12, p. 41), the use of the coupled DOT-IV calculations can be considered a reasonable alternative to a MORSE calculation. The average percent error agreement of results in this region is approximately 5% which improve to as good as 3.22% at the leg 2 boundary at detector 14. These results may be obtained at a savings of 47% of the cost of a MORSE calculation.

COUPL allows considerable flexibility as leg 2 parameters are largely independent of leg 1 parameters. Spatial mesh, directional quadrature, length, diameter, and location of the boundary source from the leg 1 centerline can be set by the user.

Although there are many related geometric coupling transformations to research (such as coupling through angles other than 90 degrees), COUPL is a reasonably accurate, convenient, and economical means of coupling successive DOT-IV cylindrical transport problems through perpendicular intersections.

LIST OF REFERENCES

LIST OF REFERENCES

1. W. A. Rhoades, D. B. Simpson, R. L. Childs, and W. W. Engle, Jr., The DOT-IV Two-Dimensional Discrete Ordinates Transport Code with Space Dependent Mesh and Quadrature, ORNL/TM-6529 (1979).
2. A. F. Henry, Nuclear Reactor Analysis, MIT Press (1975).
3. M. B. Emmett, C. E. Burgart, and T. J. Hoffman, DOMINO, A General Purpose Code for Coupling Discrete Ordinates and Monte Carlo Radiation Transport Calculations, ORNL/TM-4853 (1973).
4. Radiation Shielding Information Center Code Package CCC-203/MORSE-CG, "General Purpose Monte Carlo Multigroup Neutron and Gamma-Ray Transport Code with Combinatorial Geometry," contributed by Oak Ridge National Laboratory.
5. J. E. Aull, Acceleration of the Inner Iteration of the DOT-IV Transport Code Using a New Source Correction Scheme, ORNL/TM-7404 (1979).
6. G. I. Bell and S. Gladstone, Nuclear Reactor Theory, Van Nostrand Reinhold Company (1970).
7. R. D. Richtmeyer and K. W. Morton, Difference Methods for Initial-Value Problems, John Wiley & Sons, Inc. (1967).
8. W. Magnus and F. Oberhettinger, Formulas and Theorems for the Special Functions of Mathematical Physics, Chelsea Publishing Company (1949).

APPENDICES

APPENDIX A. FIDO INPUT

The input to COUPLE is written in the FIDO format. All data with the exception of the title card are read using the same format. The free-form version of FIDO has been slightly altered to make it machine independent except for having to change 2 cards in subroutine FFREAD. The necessary changes are indicated on comment cards.

Old Format (Fixed Form)

Each card is divided into six 12-digit data fields which are in turn divided into 3 subfields. The first subfield is a two-digit integer; the second subfield contains either a \$, *, R, I, T, S, F, A, E, Z, Q, L, N, M, U, +, -, or a blank. The third subfield contains either a fixed or floating point number. The contents of the first two subfields will define the operation to be performed on the third field.

Blank fields are ignored. One can use any or all fields on a card. For example, a box of blank cards sandwiched anywhere in a data array would be completely ignored.

Each data array is identified by a two-digit integer in a first subfield. There are both fixed and floating point arrays; a fixed point array is designated by a \$ in the second subfield, a floating point array by an *.

The second subfield contains an operator which specifies the type of operation to be performed on the data. The possible operators are listed below.

Array Operators

\$ indicates the beginning of an integer array. The first subfield contains a one- or two-digit number identifying the array.

* indicates the beginning of a floating point array. The first subfield identifies the array.

R indicates that the entry in the third subfield is to be repeated the number of times specified in the first subfield.

I indicates linear interpolation between the entry in the third subfield of the next data field. The number in the first subfield gives the number of points to be placed equally spaced in the specified range.

T indicates termination of data reading for a block. A block can contain any number of arrays. Data on a card after a T will be ignored.

S indicates skip. The first subfield defines the number of entries to be skipped. The third field can contain the first entry following the skips. A blank third subfield would be ignored.

F is used to fill the remainder of an array with the item given in the third subfield.

A is used to address a particular location in an array. This location is specified in the third subfield; the first subfield is blank.

E may be used to end specifying data for an array. This option is particularly useful when it is desired to replace only some items in a particular array. The items in question are replaced, and the use of an E prevents having to count and skip to the end of the array.

Z indicates that a zero is repeated the number of times specified in the first subfield.

Q is used to repeat sequences of numbers. The length of the sequence is defined in the third subfield. The number of times to repeat the sequence is given in the first subfield.

L is similar to I except that a logarithmic interpolation is performed between the entry points. This option is particularly useful for defining energy structures equally spaced in lethargy.

N is used to repeat a sequence of numbers in reverse order. The length of the sequence is given in the third subfield.

M is used to negate and repeat an inverted sequence. The length of the sequence is given in the third subfield.

U is used to replace the ANISN input format for an array. The array number is given in the first subfield. The format, written in normal FORTRAN, is specified on the card immediately following the card containing a U. The parentheses normally encapsulating a format should be included.

V specifies that the array identified in the first subfield will be read according to the last variable format read in.

+ or - indicates exponentiation. The + may be either a 12 punch or a 12-8-6 punch. The data in the third field is multiplied by 10^{+N} , where N is an integer in the first subfield. This option allows one to specify a number in up to nine significant digits.

Integer data in the third subfield must be adjusted. Floating point data may be written with or without an exponent. If the decimal is omitted, it is assumed to be immediately to the left of the exponent field. If there is no exponent, the decimal point is assumed to be to the extreme right of the nine-column subfield.

Input Restrictions

The following restrictions must be observed when using the FIDO input format:

- (1) Blank data fields are ignored.

(2) If the interpolation option (I) is used, the next data field may not be either blank or an A entry.

(3) The third subfield of a data field containing a \$ or an * may contain an integer, N. The next data entry is assumed to be the (N+1)th member of the array. Normally, the third subfield is blank and is ignored.

(4) All data arrays must be filled with the correct number of entries. A data array is ended by either starting a new data array or by ending a data block.

The sample data sheet (Fig. A-1) which follows illustrates possible uses of many of the FIDO input options. This is shown for illustrative purposes only and should not be construed to represent meaningful data.

New Format (Free-Form)

The transferral of input data to input forms or punched cards for a code requiring significant amounts of input is always a time consuming, distasteful and error-prone process. The original ANISN formats were designed to help reduce these difficulties. The options are convenience features. The usefulness of the "F" option which fills an array is obvious, but it is somewhat harder to see the practical uses for some of the more obscure ones like N, M, and Q; however, frequent use will turn up situations where these options are invaluable. For example, the S_n cosines are negated and reflected about 90° , a fact which suggests the use of the M option.

There are justifiable complaints with the input formats; for example, one being that data, where convenience options are not applicable, can be hard to write, because of the manner in which the data fields are spread on the card. This is especially true of integer arrays, where the data are

R - REPEAT		I - INTERPOLATE		S - SKIP		T - TERMINATE		IDENTIFICATION	REMARKS (DO NOT PUNCH)
1	1	1	1	1	1	1	1		Begin the 1\$ array
13	1	1	1	1	1	1	1		Fill the 1\$ array with 11's
26	1	1	1	1	1	1	1		Begin the 2* array
37	2	1	1	1	1	1	1		The first data item of the 2* array is 0.0
49	1	0	1	0	1	1	1		Put in 92 equally spaced points between
61	9	2	1	0	1	1	1	..	0.0 and 100.0 (1.0,2.0,3.0,....,99.0)
1	1	1	0	0	1	0	1		Repeat 25.0 four times
13	4	1	5	1	0	1	1		Skip four data locations
26	4	1	1	1	1	1	1		The next data item is 10.0
37	1	0	1	0	1	1	1		Begin 15\$ array
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1		Repeat the previous three data items
13	1	1	1	1	1	1	1		Repeat the previous 6 data items. The
26	1	1	1	1	1	1	1		15\$ array now has the sequence 1,2,3 four times
37	1	1	1	1	1	1	1	6	Terminate a data block.
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1		
13	1	1	2	3	4	1	1		
26	1	1	2	1	3	4	1	- 1	
37	1	1	2	1	3	4	1	E - 0, 1	
49	5	1	1	2	3	4	1	0 + 2	
61	3	1	1	1	1	2	3	4	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	
1	1	1	1	1	1	1	1	..	
13	1	1	1	1	1	1	1	..	
26	1	1	1	1	1	1	1	..	
37	1	1	1	1	1	1	1	..	
49	1	1	1	1	1	1	1	..	
61	1	1	1	1	1	1	1	..	

right adjusted in 12-column fields. The FIDO input forms help to some extent, but the actual keypunching is still troublesome for the layman.

The awkwardness of the input format described in the preceding paragraph has been eliminated by Ward Engle who has designed and implemented an all-FORTRAN free-form FIDO input scheme which has data items separated by blanks (as others do), but still allows all of the important convenience features of the earlier formats. The restrictions on the use of this input are essentially that the user write the data in a form that he can interpret within the context of the FIDO options. Data is easily written and keypunched, since there is no worry about which type character falls in which column or how many blanks are left between entries.

The free-form input can be interspersed with the fixed form input. To select free-form, an array is identified as either a \$\$ or an ** array, for integer and floating point arrays, respectively.

The restrictions are:

- (1) Any third subfield (data entry) must be followed by one or more blanks. This is an obvious restriction, otherwise data interpretation would be impossible.
- (2) Only columns 1-72 are used.
- (3) Numbers with exponents must not have imbedded blanks; e.g., use 1.0E+4, not 1.0 +E+4 or 1.0E+ 04.
- (4) The old + or - options (2nd subfield) are not operational. Significance requirements which led to the development of this option can be had directly.
- (5) Never enter more than 9 digits in a number. The exponent is not counted; e.g., 9234+09, 923400000+1 will work, 9234000000 will not work.

Nine-digit accuracy is clearly beyond the significance available for single precision IBM 360 floating point operations.

(6) Do not use a blank between items which fall in the first and second subfields with the old format, e.g., 24R, not 24 R. Note that the 99 restriction on the number of repeats, interpolations, etc., has been eliminated.

(7) The Z-entry must be entered as 738Z, not Z 738. The old format allowed either.

(8) the Q, M, N entries must be specified as Q4, not 4Q. The old format allows either. An entry like 3Q4 accomplishes the same as Q4 Q4 Q4. This is now true for either format.

Any character other than the digits 0 - 9, +, -, ', A, C, E, F, N, M, O, Q, T, or blank in column 1 of a card will cause the contents of the card to be listed as comments, while the data is read in. Column 2 should contain the proper carriage control character, e.g., blank, 0, 1, 2, etc. This card is ignored as a data card. This option is also available with the old formats.

Some examples of the new format are given below:

```
1$$$25R1_0_4_3Q3_2$$_3R42_E_T
```

The first 25 entries of the 1\$ array are 1's followed by 0 and 4 and then the sequence 1 0 4 is repeated three times. The 2\$ array has three 42's and then data input to the array ends. The T terminates a data block.

```
42**_0.0_0.1666667_0.3333333_N2
```

```
43**_-1.0_-0.8819171_0.3333333+M2
```

This example puts 0.0, 0.1666667, 0.3333333, 0.333333, 0.1666667 in the 42* array and -1.0, -0.8819171, -0.3333333, 0.3333333, 0.8819171 in the 43* array.

APPENDIX B. DOT-IV CARD INPUT SPECIFICATIONS

A. Title Card (72 alphameric characters).

B. Parameter Input (Description on Output Edit)

61\$	Data Set Logical Unit Reference Numbers
NTFLX	Flux Guess Input Unit (Not Used if = 0)
NTFOG	Flux Output Unit (Not Used if = 0)
NTSIG	Cross-Section Unit
NTBSI	Boundary Source Input Unit (Not Used if = 0)
(5) NTDSI	Distributed Source Input Unit (Not Used if = 0) (Must be supplied if INPSRM.GT.0)
NTFCI	First-Collision Source Input Unit (Not Used if = 0)
NTIBI	Internal Boundary Source Input Unit (Not Used if = 0)
NTIBO	Internal Boundary Source Output Unit (Not Used if = 0)
NTNPR	Large-Scale Print Unit (All print on standard output unit if NTNPR = 0)
(10) NTDIR	Directional Flux Output Unit (Not Used if = 0)
NTDSO	Distributed Source Output Unit (Not Used if = 0) (Terminate Array with "E")
62\$	Integer Control Parameters (RV=Recommended Value)
LADJ	0/1 = Forward/Adjoint Calculation
ISCTM	Maximum Order of Scattering
IZM	Number of Material Zones
IM	Maximum Number of 1st-Dimension Spatial Intervals (Negative Indicates Number of Intervals Varies with 2nd Dimension)

(5)	JM	Number of 2nd-Dimension Spatial Intervals	
	IGM	Number of Energy Groups	
	IHT	Position of Total Cross Section in Cross-Section Table (See Note 3)	
	IHS	Position of Self-Scatter Cross Sections in Cross-Section Table (See Note 3)	
	IHM	Length of Cross-Section Table for Each Group	
(10)	MIXL	Mixing Table Length (See Note 3)	
	MCR	Spare - No Effect	
	MTP	Number of Material Cross-Section Sets From NISIG (0 Implies MTP=MTM)	
	MTM	Total Number of Materials Including Mixtures	
	IDFAC	0/1 = No Density Factors/D(I,J) Input as 3*	
(15)	MM	Maximum Number of Directions in Quadrature (Negative Indicates Quadrature Specified Locally as 87\$)	
	INGEOM	0/1/2/3/4/5/6 = Geometry Option - X-Z, R-Z, R-O, 180°-360° Triangular, 60° Triangular, 90° Triangular, 120° Triangular (Also See Data Note 1)	
	IBL	Left Boundary Condition	0=Void 4=Fixed Boundary 1=Reflected Source 2=Periodic 5=Albedo 3=Cylindrical
	IBR	Right Boundary Condition	
	IBB	Boron Boundary Condition	
(20)	IBT	Top Boundary Condition	
	ISCM	Outer Iteration Maximum	
	IFXMI	Initial Inner Iteration Maximum Per Group (Negative Indicates Maximum Given as 28\$)	
	IFXMF	Find Inner Iteration Maximum Per Group (IFXMI Used if IFXMF=0)	

MODE 0/1/2/3/4/5 = Flux Extrapolation Model - Linear with
Step Fixup of Negatives/Linear with Negatives Allowed/
Step/Ordinary Weighted Difference/Linear with Super-
Weighted Fixup/Super- or θ -Weighted Difference
(RV = 0 if KTYPE=1, Otherwise 3)

(25) KTYPE 0/1/2/3/4 = Calculation Type -
Fixed Source/k-Eigenvalue/DE**2 Search/
Concentration Search/Dimension Search

IACC 0/1/2 = Rebalance Method -
Groupwise/Source Correction Method*/Space-Dependent
Scaler Method (RV=2)

KALF 0/1 = Rebalance Method - J Method/ ϕ Method
(RV=0)

IGTYPE 0/N = Discrete Ordinates Calculation/Alternate Theory
Specified by Group (7\$) For N Outers (See Note 2)

INPFXM 0/1/2/3 = Flux 0 or on NTFLX if NTFLX>0/
Flux Specified as FIJ (I,J) For each Group/FIJ(I,J)
*FG(G)/FI(I)*FJ(J)*FG(G)
(See 93*, 94*, 95* Arrays)

(30) INPSRM 0/1/2/3 = Distributed Source Input Options
Separable as for INPFXM (See 96*, 97*, 98* Arrays)

NJNTSR Interior Boundary Source at NJNTSR J-Boundaries
Input From NTIBI (May be 0)

NINTSR Interior Boundary Source at NINTSR I-Boundaries
Input From NTIBI (May be 0)

NJNTFX Interior Boundary Flux at NJNTFX J-Boundaries
Written on NTIBO (May be 0)

NINTFX Interior Boundary Flux at NJNTEFX I-Boundaries
Written on NTIBO (May be 0)

(35) LACT Number of Region and Pointwise Activities Calculated
(Negative Indicates Region Activities Only)
(See Data Note 4)

IREG Number of Regions For Activity and Balance Table Output

IPDB2 0/1/2/3=No Effect/Punch 1 DB2 Value/IGM Values
IGM*IREG Values (Negative Suppresses Balance Table Print)

IFXPRT 0/1/2=Scalar Flux Printed/Not Printed/Printed
After the Calculation of Each Group

ICSPRT 0/1=Cross-Section Data Printed/Not Printed

(40) IDIRF 0/1/2=Directional Flux Not Saved/Saved and Printed/
Saved But Not Printed (Will be Written on NTDIR if
NTDIR>0 and IDIRF>0)

JDIRF First J-Interval For Directional Flux Output

JDIRL Last J-Interval For Directional Flux Output

NBUF Number of K-Bytes Allowed For Buffer Area
(Default=60) (RV=9 on CDC)

IEPSBZ 0/1/11/21=No Effect/Use Zone Importance Convergence
(24* array)/Use and Print Convergence After Last
Inner/Use and Print Convergence on Every Inner

(45) MINBLK Minimum J-Blocking (0=All Groups in Memory Allowed)
(RV=0)

MAXBLK Maximum J-Blocking (Default=8)

ISBT I-Set for Boundary Fluxes (Default=1)

MSBT	M-Set For Boundary Fluxes (Default=1)
MSDM	M-Set For Dimensioning (Default=1)
(50) IBFSCL	Number of Flux Iterations Before First Rebalance (RV=1 or 2)
INTSCL	Number of Flux Iterations Between Rescaling (RV=1)
ITMSCL	Maximum Number of Rebalance Iterations (Default=100)
NOFIS	0/1/2=Fission with X Normalized/Fission With Input X/No Fission
IFDB2Z	0/1=No DB2/DB2 Supplied by Group, Then Region
(55) ISWP	Type of Diffusion Theory Sweep (RV=0) (See Data Note 1)
KEYJN	J-Interval For Key Flux Print (Ignored if 0)
KEYIN	I-Interval For Key Flux Print (Ignored if 0)
NSIGTP	0/1=NTSIG IS GIP Format/ORDOSW Format
NORPOS	Normalize OUTPUT To Table Position NORPOS*FLUX*VOL (0 ignored)
(60) NORMAT	Material Number to be Used in Normalization (0 Implies Use Macro Cross Section Set, Neg Implies Do Not Use Density Factor)
MSTMAX	Maximum Number of M-Sets (0 Implies JM Sets Allowed)
NEGFIX	0/1=No Effect/Negative Sources Repaired (RV=0) (Terminate Array With "E")
63*	Real Control Parameters
TMAX	Maximum Minutes of CPU Time For This Problem (0 Ignored)
XNF	Value to Which Source is Normalized (0 Ignored; Problems With KTYPE-1 MUST NOT USE 0)

	EPS	Eigenvalue Convergence Criterion on Outer Iterations
	EPP	Pointwise Flux Convergence Criterion on Inner Iterations
(5)	EPV	Volumetric Flux Convergence Criterion on Inner Iterations
	EPF	Pointwise Fission Convergence Criterion on Outer Iterations
	EKOBJ	k-Effective Sought in Search
	EVTH	k-Effective Convergence Ratio
	EVCHM	Maximum EV Change Per Iteration
(10)	EVMAX	Maximum Allowed EV or I/EV
	EVKMX	Maximum allowed K-Effective -1
	EVI	Initial Eigenvalue
	DEVDKI	Initial Eigenvalue Slope
	EVDELK	Initial Eigenvalue Increment
(15)	SORMIN	Maximum Relative Fission Extrapolation (RV=0.5 if KTYPE=1, Otherwise 0)
	CONACC	Rebalance Acceptance Criterion (Default=1.0E-3)
	CONSCL	Rebalance Convergence Objective (Default=1.0E-4)
	CONFIX	Relative Flux Threshold (RV=0)
	WSOLOI	Rebalance Outer Iteration Coefficient (RV=0)
(20)	WSOLII	Rebalance Inner Iteration Coefficient (RV=0)
	WSOLCN	Rebalance Constant (RV=1)
	ORF	Diffusion Theory Flux Acceleration (RV=0.6)
	FSNACC	Fission Density Acceleration Factor (RV=0)

FLXMIN Minimum Flux For Convergence Tests (RV=0)
 (Terminate Array With "E")

[Lengths of 61, 62, 63 arrays are subject to]
 [change from time to time.]

T

C. Primary Dimension-Setting Arrays (Omit arrays not needed).

NOTE: The Symbol "#" denotes the word "number."

71\$ ISET(J) (#=JM) [IM<0]

Index of radial mesh set to use at each J level.

Default = Set 1

72\$ IMS(ISET) (#=JM) [IM<0]

Number of intervals in each I set; then fill array with
 0's.

73\$ MMS(MSET) (#=MSTMAX) [MM<0]

Number of directions in each M set; then fill array with
 0's.

74\$ ISZNG(IG) (#=IGM) [MM<0]

Super Group Number by Group

Default = Supergroup 1

75* SZNBZ(JSZ) (#=JM) [MM<0]

J Super Mesh Boundaries

Enter as many boundaries as desired. Then fill with 0.

Default = 1. Super Zone.

76* SZNBR(ISZ) (#=|IM|) [MM<0]

I Super Mesh Boundaries

Enter as in 75*.

77\$ ISCTG(IG) (#=IGM) [ISCTM<0]

Order of scattering by group (Default = 0).

78\$ NSIG(MT) (#=MTM) [ISCTM<0]

Order of scattering by material (Default=0)

T

From these, the following are determined:

MGSZN = largest super group number [1 if MM>0]

NJSZN = number of J super zone boundaries [1 if MM>0]

NISZN = number of I super zone boundaries [1 if MM>0]

ISM = # non-zero entries in 72\$ [1 if IM>0]

MSM = # non-zero entries in 73\$ [1 if MM>0]

IMSISM = sum of IMS, all I sets [IM if IM>0]

MMSMSM = sum of MMS, all M sets [MM if MM>0]

IMSJM = sum of IMS(ISET(J)), all J [IM*JM if IM>0]

IMA = IM

MMA = MM

MMSIMS = MMA*IMA

MMSJM = MMA*JM

IHP = IHM+1 if IHS IHT+1, otherwise = IHM

D. Secondary Dimension-Setting Arrays

NOTE: The symbol "#" denotes the word "number."

81* W(N,MSET) (#=MMSMSM)

Directional weights

82* EMU(M,MSET) (#=MMSMSM)

η , cosine of angle wth X or R direction.

83* ETA(M,MSET) (# = MMSMSM)

η , cosine of angle with Y, Z, or θ direction.

84\$ IZNRG(IZ) (#=IZM) [IRED>0 OR LACT#0]

85* ZCMB(JC) (#=JM)

Enter as many boundaries as desired. Then fill with 0.

Default = 1 course mesh for each interval. Only upper

boundaries are entered; i.e., do not enter 0.0 if the

ZIN array starts with 0.0.

86* RCMB(IC) (#=IMA)

I coarse mesh boundaries, as in 85*.

87\$ IJGSZ(ISZ,JSZ,IGSZ) (#=NISZN*NJSZN*NGSZN) (MM<0]

M set by I super mesh, then by J

super mesh, then by super group

T

From these, the following are determined:

JCM = # of coarse mesh boundaries entered in the ZCMB array

ICM = # of coarse mesh boundaries entered in the RCMB array.

NREG = maximum of IZNRG(IZ) array if entered. Otherwise IZM.

E. Data Arrays (Omit arrays not needed)

NOTE: The symbol "#" denotes the word "number."

- 1* CHI(IG) fission fractions, , by group (#=IGM)
- 2* ZIN(J) Y, Z, or θ space-mesh boundaries (#=IGM)
- 3* DNIJ(I,J) density factor (#=IMSJM) [IDFAC>0]
- 4* RIN(I,ISET) X or R space-mesh boundaries (#=IMSISM+ISM)
- 5* ENER(IG) Top energy group boundaries + bottom energy
of last group + bottom energy of last neutron group,
or 0 if all groups are gammas. (IGM+2) [NTFOG>0]
- 6* DB2Z(IZ) DB^2 by group, then by region (#=IGM*NREG)
[IFDB22>0]
- 7\$ ITHYG(IG) theory by group (#=IGM) [IGTYPE>0]
(See Note 2)
- 8\$ IJZN(I,J) material zone by fine space mesh (#=IMSJM)
- 9\$ IZMT(IJZN) material number by material zone (#=IZM)
(Negative signs have no effect on this array)
- 10\$ MIXT(MIX) nuclide ID (#=MIXL) (See Note 3)
- 11\$ NUCL(MIX) nuclide ID (#=MIXL)
- 12* DENS(MIX) number density (#=MIXL)
- 13\$ MATL(MT) ID number for material (#=MTM)
(Negative for all PL components means use successive
ID's for $L>0$) (Default MATL(MT)=MT)
- 14* ZNTSR(JNTSR) Y, Z, or θ boundary positions for
J-boundary source input (#=NJNTSR)

- 15* RNTSR(INTSR) X or R boundary positions for I-boundary
 source input (#=NINTSR)
- 16* ZNTFX(JNTFX) Y, Z, or θ boundary positions for
 J-boundary flux output (#=NJNTFX)
- 17* RNTFX(INTFX) X or R boundary positions for I-boundary
 flux output (#=NINTFX)
- 18* FJSRZ(JSZN) J-Super zone search fraction (#=NJSZN)
 [KTYPE=4]
- 19* FISRZ(ISZN) I-Super zone search fraction (#=NJSZN)
 [KTYPE=4]
- 20* ABDOL(IG,J) left boundary albedo (#=IGM*JM) [IBL=5]
- 21* ABDOR(IG,J) right boundary albedo (#=IGM*JM) [IBR=5]
- 22* ABDOB(IG,I) bottom boundary albedo (#=IGM*IMA) [IBB=5]
- 23* ABDOT(IG,I) top boundary albedo (#=IGM*IMA) [IBT=5]
- 24* EPSBZ(IZ) flux error importance by material zone
 (#=IZM) [if IEJPSBZ>0]
- 25\$ ICMAT(IAC) material to be used in activity calculations
 (#=|LACT|) (See Data Note 4) [LACT#0]
- 26\$ ICPOS(IAC) cross section table position for activity
 (#=|LACT|) [LACT#0]
- 27* ACMUL(LAC) activity multiplier (#=|LACT|) [LACT#0]
- 28\$ ITMBG(IGM) initial iteration limit by group (#=IGM)
 [if IFXMI<0]

T

F. External Boundary Source Input [INGEOM<10 and IBL,IBR,IBB, or IBT=4]

NOTE: The symbol "#" denotes the word "number."

91* SII(M,J) (#=MMA*JM) [IBL=4 or IBR=4]

I boundary source for a group

92* SJI(M,I) (#=MMA*IMA) [IBB=4 or IBT=4]

T

Left and right sources are intermingled in the same array according to directions. If $\mu > 0$, the source applies to the left boundary; if $\mu < 0$, to the right. Likewise, if $\mu > 0$, a J-boundary source applies to the bottom; if $\mu < 0$, the source applies to the top boundary.

If IADJ>0, the group of lowest energy must be entered first. Omit block and delimiter if INGEOM $\geq$ 10, or if no arrays are required.

G. Flux Guess Input [INPFXM<0]

93* FIJ(I,J), FIJ(I,J) or FI(I) as INPFXM=1, 2, or 3

(#=IMSJM,IMSJM, or IM) [INPFXM>0]

94* FJ(J) (#=JM) [INPFXM=3]

95* FG(IG) (#=IGM) [INPFXM>1]

Each array of this block must be followed by "T." If no arrays are required, no "T" is required. If LADJ>0, the group of lowest energy must be specified first.

H. Distributed Source Input [INPSRM>0]

96* as 93*

97* as 94*

98* as 95*

Notes:

(1) Special Geometry Features

If INGEOM $\geq$ 10, a one-dimensional problem will be solved with vertical flow suppressed. Use void boundary condition at top and bottom. No vertical leakage will result.

If INGEOM=10 or 11, slab or cylindrical geometry will result, with sources specified in the normal way. If JM>1, the effect is, in general, to run several independent one-dimensional problems. A k-calculation of this type would be difficult to interpret, however.

If INGEOM=20, KTYPE=0, and IBR=4, a combined reflection/transmission problem is solved. Omit the 91* and 92* array blocks, and supply the input spectrum as CHI. The code will generate a boundary source in the leftmost direction of non-zero weight of each downward η level. The value of JM must be the number of downward η levels. In this case, J corresponds to the incident η level, not to a Y-direction mesh. The 2* array should be filled with the values 0.0, 1.0, 2.0, ... JM. The first η level will correspond to the last J level. $|CHI(IGM)|$ particles will enter each level in each group. If CHI(I)<0, the emerging flux is printed by direction, then by space interval.

If INFWOM=30, emerging flux will be calculated for a source entering each group JDIRF $\leq$ IG $\leq$ JDIRL and leaving all groups IGI $\leq$ IGO $\leq$ IGM. The total source per group in each J-level is $|CHI(IGI)|$. If CHI<0, the flux from sources entering that group will be printed. The problem is "double differential," in that the flux leaving each group corresponding to a source in each input group is calculated.

If $3 \leq \text{INGEOM} \leq 6$, and if all iterations are to be performed in diffusion theory, equilateral-triangular geometry is available. Geometry options, together with the required value of IM, are:

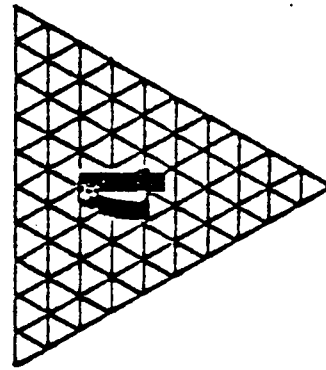
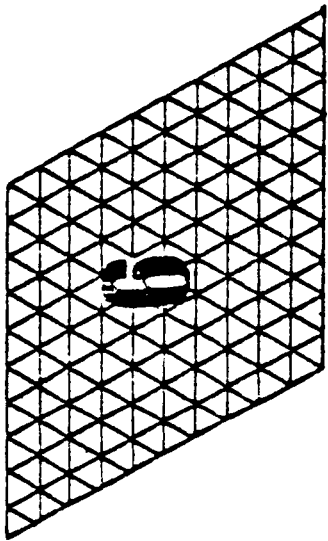
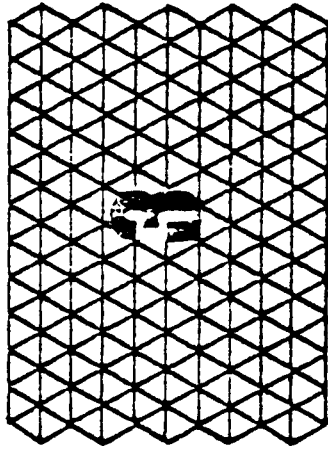
<u>INGEOM</u>	<u>TRIANGULAR GEOMETRY OPTION</u>	<u>VALUE OF IM</u>
3	180°-360° Symmetry	User's option
4	60° Symmetry	2*JM-1
5 (not oper)	90° Symmetry	2*JM+1
6	120° Symmetry	2*JM

The resulting geometry is illustrated in Figure 5.1.

The mesh input data involve certain unusual requirements. In most cases, the value of IM is a function of JM as shown above. Also, the radial dimensions used for input do not correspond directly to the actual dimensions of the mesh. For all the 90° symmetry, every mesh interval is an equilateral triangle with sides having a length s .

JM+1 J-interval boundaries are entered, beginning with 0, and with spacing $\sqrt{3} S/2$. With 120°, 180°, or 360° symmetry, IM+1 I-interval boundaries are entered, beginning with 0, and with spacing $S/2$. With 60° symmetry, the code will automatically set IM to negative, and arrange for the entry of JM I-sets. The j th I-set has $2j$ boundaries with spacing $S/2$. The description applicable to 90° symmetry is not available.

For all cases, the (1,1) mesh interval is a triangle with vertex down. Example meshes are shown for the various triangular geometry options in Fig. B-1.



INGEOM	SYMMETRY
3	180-360
4	60
6	120

Fig. B-1. Triangular Geometry Options.

For 60° symmetry, the parameter ISBT should be set equal to JM, and the bottom boundary condition must be reflected. For 120° symmetry, the left and bottom boundaries must be periodic if periodic boundary conditions are selected.

(2) Special Theory Options

If IGTYP > 0, then the first IGTYP outer iterations may be performed using alternate theories specified by group in the 7\$ array. If the gth entry is N_g , then the first $|N_g|$ inner iterations of the first IGTYP outer iterations on group g will be performed using diffusion theory if $N_g > 0$ and combined P_1 theory if $N_g < 0$. If $|N_g|$ is less than the applicable maximum number of iterations, the iterations will be completed in transport theory, even though the alternate theory may have achieved convergence.

If diffusion theory is chosen, values for ISWP and ORF must be entered:

ISWP

METHOD OF SWEEP

- 0 Line inversion in direction of largest number of mesh intervals
- 1 Alternating direction (line-column) on consecutive outer iterations
- 2 Line inversion
- 3 Column inversion
- 4 Alternating direction (line-column) on consecutive inner iterations
- 5 Alternating direction (column-line) on consecutive inner iterations
- 6 Line inversion outward from center J interval

ORF Flux acceleration parameter $0 \leq \text{ORF} \leq 1$ (RV = 0.5)

If combined P_1 theory is chosen, the code will automatically revert to row-scored mode if $MAXBLK > 0$. Otherwise, an error stop will result. Also, ISCT must be 1 if P_1 is chosen.

(3) Cross Section Input and Mixing

The internal cross section storage comprises MTM sets of cross sections for each energy group. The first MTP sets are read from logical unit NTSIG, which is always required. The remainder are prepared using a "mixing table," described below. It is intended that the first MTP "materials" will be microscopic nuclide data, and the remainder will be macroscopic mixtures, although other uses are possible. The material numbers entered in the IXMT array (9\$) must be integers between 1 and MTM, corresponding to the appropriate data.

Each set of data consists of IHP cross sections as indicated in the description of the ORDOSW input file. If $ISXTM > 0$, the ISCTM sets of Legendre expansion data must follow each set designated in the IZMT array. If $ISCTM > 0$, then the required number of expansion sets is indicated in the NSIG array (78\$).

If $MIXL > 0$, cross sections are to be modified by a mixing table, specified by the MIXT, NUCL, DENS, and MATL arrays (10\$, 11\$, 12*, and 13\$). The MATL array assigns an arbitrary ID number to each material. If it is not entered the ID's 1,2,...., MTM are assumed. Each integer entered in the MIXT and NUCL arrays must be one of the ID's, or 0. In the following:

m = number of the material having ID $|p|$ if $|p| > 0$, else 0

n = number of the material having ID $|q|$ if $|q| > 0$, else 0

and the interpretation of the table is as follows:

MIXTNUCLDENS

p

q

d

- (a) If $q=0$, then the data of material m will be multiplied by d .
- (b) If $q>0$, then the data of material n , multiplied by d , will be added to that of material m .
- (c) If $q=p$, the data material m will be multiplied by the eigenvalue in concentration searches.
- (d) If $p<0$, the appropriate Legendre expansion components will be treated as was the principle set.

Each set of entries is executed in sequence. In searches, the table may be executed repeatedly. As an example, with $MIXL=6$, $MTM=11$, and $ISCTM=3$.

<u>I</u>	<u>MIXT</u>	<u>NUCL</u>	<u>DENS</u>	<u>MATL</u>
1	1000	0	0	10
2	1000	10	.1	20
3	1000	20	.2	30
4	1000	1000	0	40
5	-2000	0	0	50
6	-2000	30	.3	60
7				1000
8				2000
9				3000
10				4000
11				5000

Material 7 will consist of:

Material 1 * .1 + material 2 * .2 and will be multiplied by the eigenvalue. Material 8, 9, 10, and 11 will be material 3, 4, 5, and 6 * .3.

Although the value of σ^A does not affect the flux in a calculation directly, it must be used to obtain correct balance tables. It should meet the condition

$$\sigma_g^T = \sum_g^A + \sum_{g'} \sigma_{g g'}$$

If σ_g^T is replaced by a "transport cross section" in reactor core problems, $\sigma_{g \rightarrow g}$ must be reduced such as to maintain σ^A constant.

It may be important to note that many standard cross section files such as ISOTXS and MATXS have Legendre expansion data which must be multiplied by $2\ell+1$, where ℓ is the expansion index, for use in DOT-IV. It is intended that this be done in the code which prepares the ORDOSW (or optional GIP input file).

(4) Activity Edits

A very general provision for obtaining energy-integrated reaction rates is provided. The arrays ICMAT, ICPOS, and ACMUL define an "activity table," interpreted as follows:

	<u>ICMAT</u>	<u>ICPOS</u>	<u>ACMUL</u>
	r	s	t
$A_{i,j} = t_j \sum_i d_{i,j}$	$\sum_i \sum_j d_{i,j}$	$C_{i,j,r,s}$	$\sum_g \phi_{g,j,i} \sigma_{g,r,s}$
$B_z =$	$\sum_{i,j} X_{i,j,z}$	$A_{i,j}$	$V_{i,j}$

where

i, j = space mesh indices

g = energy group index

$|r|$ = material number

$|s|$ = cross section table position number

t = arbitrary multiplier

d = density factor if used, else 1

ϕ = flux

σ = cross section

C = the "number density" of material $|r|$ in space cell i, j

$X = 1$ if cell i, j is in edit region Z , else 0

The value of C is determined as follows:

- (a) If r is the macro material used in cell i, j , then $C = 1$,
- (b) If r is 0, then $C = 1$, and the cross sections for the macro material in cell i, j are used,
- (c) If $r < 0$, then $C = 1$,
- (d) If $r > 0$ and is not the macro, then C is the sum of all assignments or r to the macro according to the mixing table. Suppose $LACT=4$ and the following table is entered, referring back to the mixing table illustrated above; and suppose material 7 is named as a macro in a portion of the problem:

<u>I</u>	<u>ICMAT</u>	<u>ICPOS</u>	<u>ACMUL</u>
1	7	3	1.
2	-7	3	1.
3	0	4	500.
4	1	5	1.

Activity 1 will use position 3 of material 7, but will be 0 where 7 is not the macro. Activity 2 will be similar, but will be calculated whether 7 is the macro or not. Activity 3 will use position 4 of the macro in each space cell, and will multiply the result by 500. Activity 4 will use position 5 of material 1, a "number density" of .1 wherever 7 is the macro.

As a special feature, if $|s| > \text{IHP}$, then the value of $\text{CHI} (1^*)$ for the group will be used to replace σ , and c and d will be everywhere 0. This can be useful in adjoint problems.

If $s < 0$, the activity will be written on the punch data set. This can be used for fission density output, for example.

APPENDIX C. GIP INPUT CARD SPECIFICATIONS

A. Title Card (72-character description)

B. Parameter Input

1\$\$ Array

IGM Number of groups

IHT Sigma(T) position

IHS Sigma(G--G) position

IHM Table length

MS Mixing Table Length

MCR Number of materials from cards

MTP Number of materials from library tape

MTM Number of materials (MCR+MTP+mixtures)

ITH 0=direct solution; 1=adjoint

ISCT Order of scattering expansion

IPRT 0=print all; 1=no print; 2=print mixtures

IOUT 0=outout all materials; 1=none; 2=mixtures only

IDOT 0=ANISN output; 1=DOT III; 2=DOT IV

NBUF No. of k-bytes buffer space (default=60)

E

T

10\$\$,11\$\$,12** arrays - mixing table(MS entries each)

7\$\$ array - mix code(optional-MTM entries)

13\$\$ array - nuclide I.D. numbers(MTP entries)

T

14** array - cross sections for nuclide 1, ANISN Format, all groups

.

.

.

(Each new nuclide must start on a new card)

.

.

14** array - cross sections for nuclide MCR, ANISN format, all groups

T

APPENDIX D. COUPL CARD INPUT SPECIFICATIONS

A. Title Card (72 alphameric characters)

B. Parameter input and Data Arrays

1\$\$ Integer Parameters

ITYP Problem type: 1=perpendicular intersecting cylinders;
 2=parallel intersecting cylinders.
 (Note: As COUPL has only been tested for the case
 of perpendicular intersecting cylinders, ITYP
 should always be set equal to 1. The ITYP=2 optio
 is not fully operational and may produce error
 messages.

NT DOT-IV flux moment input unit.

NTW COUPL angular flux output unit.

ND Disc transfer unit number for input flux moments.

MM2 Number of directions in Leg 2 quadrature.

IM1 Number of radial intervals in Leg 1.

IM2 Number of radial intervals in Leg 2.

JM1 Number of axial intervals in Leg 1.

NOT Number of azimuthal intervals for averaging boundary
 source.

IGM Number of energy groups.

2** Real Parameters

H2 Leg 1 z coordinate(height above the Leg 1 origin) of
the point $r=0$ on the Leg 2 boundary source.

RDC Radial location in Leg 1 of the point $r=0$ on the Leg 2 boundary source.

BSZ The Leg 2 z coordinate of the point $r=0$ on the Leg 2 boundary source.

OT

- 3** Leg 1 axial mesh: $H(j), j=0,1,2,\dots, JM.$
- 4** Leg 1 radial mesh: $R1(i), i=0,1,2,\dots, IM1.$
- 5** Leg 2 radial mesh: $R2(i), i=0,1,2,\dots, IM2.$
- 6** Leg 2 azimuthal mesh: $THET(n), n=0,1,2,\dots, NOT.$
- 7** 'Mu' cosine of angle with r direction in Leg 2:
 $MU(m), m=0,1,2,\dots, MM2.$
- 8** 'Eta' cosine of angle with z direction in Leg 2:
 $ETA(m), m=0,1,2,\dots, MM2.$

T

APPENDIX E. PROGRAM ACCESS

To run COUPL, one must have access to various locally standardized I/O subroutines. This can be accomplished by merely link-editing to any load modules containing these subroutines. All of the required subroutines, for example, are contained in the load modules QSAM and GIP which are members of the partitioned dataset PROG which is currently kept on-line at ORNL. These load modules provide the I/O subroutines used by COUPL. Since JCL requirements vary from time to time and from system to system, the job deck listing given below is merely an example of how COUPL was run on the X-10 3330-1 for test-phase problems.

```

//C   EXEC   FORTQC,PARM.FORT='XREF'
      COMMON LIP,LRP,LH,LR1,LR2,LTHET,LEMU2,LETA2,
      &LPHI,LMOM,LSHM,LSHMN,LPN,LA,LFLUM,LSDF,LDFLU,
      &LAST,ITYP,NT,NTW,ND,MM2,IM1,IM2,JM1,NOT,IGM,H2,RDC,BSZ,
      &DUMMY(20000)

C
      DIMENSION D(1),IFC(10),AEMUP(1),AETAP(1),AEMUM(1),AETAM(1),
      &WF(1)
      COMMON /COMIN/ TITL(12),TIMAR(10),MULT,LLKAR,NIN,NOU,NPU,
      &NPR,JERR,ISIZE,NBYTE,NSEV,NCND,NCNDGO
      EQUIVALENCE (D(1),LIP)
      READ(5,1000) TITL
1000  FORMAT(12A4)
      WRITE(6,2000) TITL
2000  FORMAT('1',6X,12A4)
C
      LIM=20000
      LIP=19
      LRP=29
      LH=32
      CALL FIDOS(1,NER,5,6,LIP,LIP,LIP)
C  READ IN FILE CONTROL PARAMETERS
      CALL DOPC(0,0,0,JERR,D(1),0,0,0)
      CALL SEQIO(0,NT,0,0,0)
      CALL SEQIO(1,NT,IFC(1),10,3)
C  CHECK KEY LEG1 INTEGER CONTROL PARAMETERS.
      IF(IFC(1).LT.IGM) GO TO 91
      IF(IFC(3).NE.JM1) GO TO 91
      IF(IFC(5).NE.IM1) GO TO 91
C  LMS IS THE LENGTH OF THE MOMENT EXPANSION.
      LMS=IFC(4)
      WRITE(6,2000) TITL
      WRITE(6,3000) ITYP,NT,NTW,ND,MM2,IM1,IM2,JM1,NOT,IGM,LMS
3000  FORMAT(////,7X,'ITYP=',I7,4X,'PROBLEM TYPE: 1=PERPENDICULAR'
      &' CYLINDERS, 2=PARALLEL CYLINDERS',/,7X,'NT=',I9,4X,'LOGICAL',
      &' UNIT NUMBER OF VARFLM DOT TAPE',/,7X,'NTW=',I8,4X,'LOGICAL',
      &' UNIT NUMBER OF BNDYRS DOT TAPE',/,7X,'ND=',I9,4X,'LOGICAL',
      &' UNIT NUMBER OF DISK DATA FROM VARFLM TAPE',/,7X
      &,'MM2=',I8,4X,'NUMBER OF DIRECTIONS IN LEG 2 QUAD',/,7X,'IM1='
      &,'I8,4X,'NUMBER OF RADIAL INTERVALS IN LEG 1',/,7X,'IM2='
      &,'I8,4X,'NUMBER OF RADIAL INTERVALS IN LEG 2',/,7X,
      &'JM1=',I8,4X,'NUMBER OF AXIAL INTERVALS IN LEG 1',/,7X,
      &'NOT=',I8,4X,'NUMBER OF LEG 2 AZIMUTHAL ANGLE INTERVALS FOR'
      &,' AVERAGING BOUNDARY SOURCE',/,7X,'IGM=',I8,4X,'NUMBER OF',
      &' ENERGY GROUPS',/,7X,'LMS=',I8,4X,'LENGTH OF MOMENT EXPANSION')
C  DEFINE BASE LOCATIONS FOR INPUT ARRAYS
      LR1=LH+JM1+1
      LR2=LR1+IM1+1
      LTHET=LR2+IM2+1
      LEMU2=LTHET+NOT+1
      LETA2=LEMU2+MM2
      LPHI=LETA2+MM2

```

```

CALL FIDOS(1,NERA,5,6,LIP,LIP,LIP)
NER=NER+NERA
IF(NER.NE.0) GO TO 99
WRITE(6,4000) H2,RDC,BSZ
4000  FORMAT(/,7X,'H2=',4X,E11.4,6X,'LEG 1 Z COORDINATE(HEIGHT',
&' ABOVE LEG 1 ORIGIN)OF THE POINT R=0 ON THE LEG 2 BOUNDARY',
&' SOURCE',/,7X,'RDC=',3X,E11.4,6X,'RADIAL LOCATION IN LEG',
&' 1 OF THE POINT R=0 ON THE LEG 2 BOUNDARY SOURCE',
&/,7X,'BSZ=',3X,E11.4,6X,'THE LEG 2 Z COORDINATE OF ',
&'THE LEG 2 BOUNDARY SOURCE')
C
WRITE(6,2000) TITL
WRITE(6,5000)
5000  FORMAT(//////,9X,'LEG#1 R',4X,'LEG#1 Z',6X,'LEG#2 R',
&4X,'LEG#2 THETA',/)
CALL WOT10(D(LR1),IM1+1,D(LH),JM1+1,D(LR2),IM2+1,D(LTHET),
&NOT+1,0,0,0,0,0,0,0,0,0,0,0,6)
WRITE(6,2000) TITL
WRITE(6,5500)
5500  FORMAT(//////,8X,'LEG#2 ETA',3X,'LEG#2 MU',/)
CALL WOT10(D(LETA2),MM2,D(LEMU2),MM2,0,0,0,0,0,0,0,0,0,0,0,0,
&0,0,0,0,6)
IF((ITYP.EQ.1).OR.(ITYP.EQ.2)) GO TO 10
WRITE(6,6000)
6000  FORMAT('1',6X,' ***ERROR***',3X,'DATA FOR ITYP NOT EQUAL TO',
&' 1 OR 2')
STOP
C DEFINE REGION IN LEG 1 ARRAYS TO BE SCANNED
10    IF(ITYP.EQ.2) GO TO 15
CALL BSET1(D(LH),D(LR1),D(LR2),JM1,IM1,IM2,H2,RDC,MINZ,MAXZ,
&MINR,MAXR)
GO TO 16
15    CALL BSET2(D(LH),D(LR1),D(LR2),JM1,IM1,IM2,HR,RDC,NZ,MINR,
&MAXR)
C THE FOLLOWING LOOP DETERMINES THE ORDER OF SCATTER
C CORRESPONDING TO THE LENGTH OF THE MOMENT EXPANSION.
16    ISCT=0
DO 17 L=1,LMS
ISCT=ISCT+1
IF(ISCT*(ISCT+3).EQ.2*(LMS-1)) GO TO 20
17    CONTINUE
C
C DEFINE BASE LOCATIONS FOR CALC ARRAYS
C
20    LMOM=LPHI+NOT+1
LSHM=LMOM+LMS*(NOT+1)
LSHMN=LSHM+LMS+1
LPN=LSHMN+LMS+1
LA=LPN+(ISCT+1)*(ISCT+1)
LFLUM=LA+2*(ISCT+1)
LSDF=LFLUM+IM1*LMS
LDFLU=LSDF+MM2
LAST=LDFLU+MM2*IM2

```

```

      ITOT=LAST+IM1*LMS
      WRITE(6,7000) ITOT,LIM
7000  FORMAT(////,6X,I7,' WORDS OF STORAGE REQUIRED FOR INPUT, VS',
      &' AVAILABLE',I7,///)
      IF(ITOT.GE.LIM) GO TO 9
C   THE NEXT TWO VARIABLES ARE NEEDED FOR BLKIO IN LOCAT
      NRECMX=JMI
      NWBLKM=0
      READ(NT)
      READ(NT)
      IREC=0
      NVAL=IM1*LMS
      PI=3.1415927
      IM2PI=IM2+1
      NOTPI=NOT+1
      DO 100 IGPI=1,IGM
C   TRANSFER TAPE DATA TO DISK UNIT #91 FOR RANDOM ACCESS
      DO 25 J=1,JMI          CALL SEQIO(1,NT,D(LAST),NVAL,IREC)
      CALL BLKIO(2,ND,D(LAST),NVAL,0,D(1),JMI,0)
25  CONTINUE
      CALL BLKIO(0,ND,0,0,0,D(1),JMI,0)
      IG=IGPI-1
      DO 90 IC=1,IM2PI
      I=IC-1
      DO 40 JC=1,NOTPI
      J=JC-1
C
C   THE FOLLOWING 6 EQUATIONS DETERMINE THE 'EXACT' COORDINATES
C   AND AZIMUTHAL ANGLE PHI IN THE LEG 1 SYSTEM OF THE CURRENT
C   (R,THETA)=(IC,JC) POINT ON THE LEG 2 BOUNDARY SOURCE.
C
      IF(ITYP.EQ.2) GO TO 30
      RE=SQRT((D(LR2+I)*SIN(D(LTHET+J)))**2+RDC**2)
      ZE=H2+D(LR2+I)*COS(D(LTHET+J))
      IF(RDC.EQ.0.0) GO TO 28
      D(LPHI+J)=ATAN(D(LR2+I)*SIN(D(LTHET+J))/RDC)
      GO TO 35
28  D(LPHI+J)=PI/2.0
      GO TO 35
30  RE=SQRT(RDC**2+D(LR2+I)**2-2.0*RDC*D(LR2+I)*COS(D(LTHET+J)))
      ZE=H2
      IF((RDC-D(LR2+I)*COS(D(LTHET+J))).EQ.0.0) GO TO 32
      D(LPHI+J)=ATAN(D(LR2+I)*SIN(D(LTHET+J))/(RDC-D(LR2+I)*
      &COS(D(LTHET+J))))
      GO TO 35
32  D(LPHI+J)=PI/2.0
C
C   LCMOM IS THE LOCATION IN D OF THE JTH SET OF MOMENTS OF
C   LENGTH LMS FOR CURRENT I (NEEDED IN LOCAT).
C
35  LCMOM=LMOM+J*LMS
C   SUBROUTINE LOCAT SCANS THE LEG 1 SYSTEM FOR THE INTERVAL

```

```

C CONTAINING THE 'EXACT' TRANSFORMATION OF THE CURRENT LEG 2
C BOUNDARY SOURCE POINT (RE&ZE) FOUND ABOVE. THE DOT MOMENT SET
C CORRESPONDING TO THIS INTERVAL IS ASSIGNED TO THE CURRENT
C LEG 2 BOUNDARY SOURCE POINT.
      CALL LOCAT(D(1),D(LR1),D(LH),D(LCMOM),D(LFLUM),RE,ZE,MINR,
      &MAXR,MINZ,MAXZ,NZ,ITYP,IM1,JM1,LMS,NVAL,NRECMX,NWBLKM,ND)
      IF(I.EQ.0) GO TO 41
40  CONTINUE
C
C AT EACH LEG 2 AZIMUTHAL ANGLE FOR THE CURRENT R BOUNDARY
C ON THE SOURCE SURFACE, THE FOLLOWING LOOP ASSIGNS A CORRESPONDING
C FLUX TO THE CURRENT LEG 2 DIRECTION AND AVERAGES ACCROSS THE
C AZIMUTHAL INTERVAL FROM THE PREVIOUS ANGLE. FINALLY, IT AVERAGES
C OVER ALL AZIMUTHAL INTERVALS USING THE INTERVAL ARC LENGTHS
C AS WEIGHTS TO OBTAIN A DIRECTIONAL FLUX FOR THE CURRENT R
C BOUNDARY ON THE SOURCE SURFACE.
C
41  DO 80 MC=1,MM2
      M=MC-1
C DETERMINE THIRD DIRECTION TO USE IN SUBROUTINE TFORM.
      XI=SQRT(ABS(1.0-(D(LEMU2+M)**2+D(LETA2+M)**2)))
C ZERO R-BOUNDARY FLUX FOR CURRENT R AND DIRECTION.
      BDFLU=0.0
      DO 60 JC=1,NOTP1
        J=JC-1
C ZERO AZIMUTHAL POINT FLUX FOR EACH AZIMUTHAL POINT.
        AZPFLU=0.0
CC SUBROUTINE TFORM TRANSFORMS THE CURRENT DIRECTION PAIR
C (MU,ETA,+XI) & (MU,ETA,-XI) INTO THE LEG 1 COORDINATE SYSTEM
C PAIR (MU,ETA,XI)+ & (MU,ETA,XI)-. NOTE THAT BOTH +XI & -XI
C NEED TO BE CONSIDERED BECAUSE IN GENERAL THEY ARE
C TRANSFORMED TO TWO CORRESPONDING LEG 1 DIRECTIONS WITH
C DIFFERING MU'S AND ETA'S WHICH REPRESENT DIFFERENT FLUX VALUES.
C THIS PAIR OF FLUXES IS LATER AVERAGED INTO ONE FLUX WHICH IS
C ASSIGNED TO BOTH (MU,ETA,+XI) & (MU,ETA,-XI) LEG 2 DIRECTIONS.
      IF(ITYP.EQ.2) GO TO 42
      CALL TFORM1(D(LEMU2+M),D(LETA2+M),XI,D(LPHI+J),D(LTHET+J),
      &EMUP,EMUM,ETAP,ETAM)
      GO TO 46
42  CALL TFORM2(D(LEMU2+M),D(LETA2+M),XI,D(LPHI+J),D(LTHET+J),
      &EMUP,EMUM,ETAP,ETAM)
C DEFINE ARRAYS FOR TRANSFORMED DIRECTION PAIR FOR PCON.
46  AEMUP(1)=EMUP
      AETAP(1)=ETAP
      AEMUM(1)=EMUM
      AETAM(1)=ETAM
      WF(1)=1.0
C
C SUBROUTINE PCON GENERATES A SET OF ASSOCIATED LEGENDRE
C POLYNOMIALS WHICH DEPEND ONLY UPON MU AND ETA (UNIT DIRECTION VECTOR).
C THESE ARE LATER EXPANDED WITH THE MOMENT SET CORRESPONDING
C TO THE CURRENT SPATIAL INTERVAL IN THE LEG 2

```

```

C MESH.
  CALL PCON(AEMUP,AETAP,WF,D(LSHM+1),D(LPN),D(LA),CP,1,ISCT+1,
    &ISCT)
  CALL PCON(AEMUM,AETAM,WF,D(LSHMN+1),D(LPN),D(LA),CP,1,ISCT+1,
    &ISCT)
C
C EXPAND MOMENTS AND ASSOCIATED LEGENDRE POLYNOMIALS TO FORM
C DIRECTIONAL FLUX
  LCMOM=L MOM+J*LMS
  DO 50 LC=1,LMS
    L=LC-1
    IF(L.NE.0) GO TO 48
    D(LSHM+L)=1.0
    GO TO 49
48  D(LSHM+L)=(D(LSHM+L)+D(LSHMN+L))/2.0
49  AZPFLU=AZPFLU+D(LCMOM+L)*D(LSHM+L)
50  CONTINUE
C
  IF(J.EQ.0) GO TO 55
  AZIFLU=(SAZPFL+AZPFLU)/2.0
  BDFLU=BDFLU+(D(LTHET+J)-D(LTHET+J-1))*AZIFLU/PI
55  SAZPFL=AZPFLU
  IF(I.NE.0) GO TO 60
  BDFLU=AZPFLU
  GO TO 70
60  CONTINUE
  DFLU=(D(LSDF+M)+BDFLU)/2.0
  D(LDFLU+(I-1)*MM2+M)=DFLU
70  D(LSDF+M)=BDFLU
80  CONTINUE
90  CONTINUE
  CALL WRBNDI(BSZ,0,D(LDFLU),IGM,JM2,IM2,MM2,0,1,IGP1,NTW)
100 CONTINUE
  CALL DOPC(4,0,0,JERR,0,0,0,0)
  GO TO 999
9  WRITE(6,9000)
9000 FORMAT('1',6X,'***ERROR***',3X,'MAXIMUM STORAGE EXCEEDED')
  GO TO 99991  NER=NER+1
99  WRITE(6,9001) NER
9001 FORMAT('1','PROBLEM TERMINATED.',I5,' INPUT ERRORS.')
999 STOP
  END
  SUBROUTINE LOCAT(D,R1,Z,MOM,FLUM,RE,ZE,MINR,MAXR,MINZ,
    &MAXZ,NZ,ITYP,IM1,JM1,LMS,IRL,NRECMX,NWBLKM,ND)
  REAL MOM(1)
  DIMENSION D(1),R1(1),Z(1),FLUM(1)
  NR=0
  IF(MINR.NE.MAXR) GO TO 5
  NR=MAXR
  IF(MAXR.EQ.1) INTR=1
  IF(MAXR.EQ.(IM1+1)) INTR=IM1
  GO TO 15

```

```

5     MAXRM1=MAXR-1
      DO 10 I=MINR,MAXRM1
        IF(ABS(R1(I)-RE).LE.ABS(R1(I+1)-RE)) NR=I
        IF(NR.NE.0) GO TO 15
10    CONTINUE
      NR=MAXR
      IF(MAXR.EQ.(IM1+1)) GO TO 17
15    IF(R1(NR).GT.RE) GO TO 17
      INTR=NR
      GO TO 18
17    INTR=NR-1
18    IF(ITYP.EQ.2) GO TO 25
      NZ=0
      IF(MINZ.NE.MAXZ) GO TO 19
      NZ=MAXZ
      IF(MAXZ.EQ.1) INTZ=1
      IF(MAXZ.EQ.(JM1+1)) INTZ=JM1
      GO TO 25
19    MAXZM1=MAXZ-1
      DO 20 J=MINZ,MAXZM1
        IF(ABS(Z(J)-ZE).LE.ABS(Z(J+1)-ZE)) NZ=J
        IF(NZ.NE.0) GO TO 25
20    CONTINUE
      NZ=MAXZ
      IF(MAXZ.EQ.(JM1+1)) GO TO 27
25    IF(Z(NZ).GT.ZE) GO TO 27
      INTZ=NZ
      GO TO 30
27    INTZ=NZ-1
30    ICRN=INTZ
      CALL BLKIO(1,ND,FLUM(1),IRL,ICRN,D(1),NRECMX,NWBLKM)
      CALL BLKIO(0,ND,0,0,0,D(1),NRECMX,NWBLKM)
      DO 35 LC=1,LMS
        L=LC-1
        MOM(LC)=FLUM(L*IM1+INTR)
35    CONTINUE
      RETURN
      END
      SUBROUTINE BSET1(Z,R1,R2,JM1,IM1,IM2,H2,RDC,MINZ,MAXZ,MINR,
&MAXR)
      DIMENSION Z(1),R1(1),R2(1)
      ZLO=H2-R2(IM2+1)
      ZHI=H2+R2(IM2+1)
      JUP=JM1+1
      RLO=RDC
      RHI=SQRT(RDC**2+R2(IM2+1)**2)
      IUP=IM1+1
      IF((ZHI.GE.Z(1)).AND.(ZLO.LE.Z(JUP))) GO TO 2      WRITE(6,100)
100  FORMAT('1',6X,' ***ERROR***',3X,'LEG 2 BOUNDARY SOURCE IS',
&' OUTSIDE THE LEG 1 AXIAL MESH')
      STOP
2     IF((RHI.GE.R1(1)).AND.(RLO.LE.R1(IUP))) GO TO 5

```

```

WRITE(6,300)
300  FORMAT('1',6X,' ***ERROR***',3X,'LEG 2 BOUNDARY SOURCE IS',
&' OUTSIDE THE LEG 1 RADIAL MESH')
STOP
5   MINZ=0
    MAXZ=0
    DO 10 J=2,JUP
      IF((Z(J).GE.ZLO).AND.(MINZ.EQ.0)) MINZ=J-1
      IF(MINZ.EQ.(J-1).AND.(ABS(Z(J-1)-ZLO).GT.ABS(Z(J)-ZLO)))
&MINZ=J
      IF(MINZ.NE.0) GO TO 15
10  CONTINUE
15  IF(ZHI.GE.Z(JUP)) MAXZ=JUP
    IF(MAXZ.NE.0) GO TO 25
    MINZP1=MINZ+1
    DO 20 J=MINZP1,JUP
      IF((Z(J).GE.ZHI).AND.(MAXZ.EQ.0)) MAXZ=J
      IF((MAXZ.EQ.J).AND.(ABS(Z(J)-ZHI).GT.ABS(Z(J-1)-ZHI)))
&MAXZ=J-1
      IF(MAXZ.NE.0) GO TO 25
20  CONTINUE
    IF(MAXZ.NE.0) GO TO 25
    STOP
25  MINR=0
    MAXR=0
    DO 35 I=2,IUP
      IF((R1(I).GE.RLO).AND.(MINR.EQ.0)) MINR=I-1
      IF((MINR.EQ.I-1).AND.(ABS(R1(I-1)-RLO).GT.ABS(R1(I)-RLO)))
&MINR=I
      IF(MINR.NE.0) GO TO 40
35  CONTINUE
40  IF(RHI.GE.R1(IUP))MAXR=IUP
    IF(MAXR.NE.0) GO TO 50
    MINRP1=MINR+1
    DO 45 I=MINRP1,IUP
      IF((R1(I).GE.RHI).AND.(MAXR.EQ.0)) MAXR=I
      IF((MAXR.EQ.I).AND.(ABS(R1(I)-RHI).GT.ABS(R1(I-1)-RHI)))
&MAXR=I-1
      IF(MAXR.NE.0) GO TO 50
45  CONTINUE
    IF(MAXR.NE.0) GO TO 50
50  WRITE(6,500) R1(MINR),R1(MAXR),Z(MINZ),Z(MAXZ)
500  FORMAT('//,8X,'THE BOUNDARIES FOR THE REGION TO BE SCANNED:',//,
&6X,'MINIMUM R= ',E12.5,2X,'MAXIMUM R= ',E12.5,/,6X,'MINIMUM Z= ',
&E12.5,2X,'MAXIMUM Z= ',E12.5)
    RETURN
    END
    SUBROUTINE BSET2(Z,R1,R2,JM1,IM1,IM2,HR,RDC,NZ,MINZ,MAXZ,MINR,
&MAXR)
    DIMENSION Z(1),R1(1),R2(1)
    NZ=0
    JUP=JM1+1

```

```

      RLO=RDC-R2(IM2+1)
      RHI=RDC+R2(IM2+1)
      IUP=IM1+1
      IF((H2.GE.Z(1)).AND.(H2.LE.Z(JUP))) GO TO 5
      WRITE(6,100)
100  FORMAT('1',6X,' ***ERROR***',3X,'LEG 2 BOUNDARY SOURCE IS OUTSIDE'
      &,' LEG 1 AXIAL MESH')
      STOP
5     IF((RHI.GE.R1(1)).AND.(RLO.LE.R1(IUP))) GO TO 10
      WRITE(6,300)
300  FORMAT('1',6X,' ***ERROR***',3X,'LEG 2 BOUNDARY SOURCE'
      &,' IS OUTSIDE LEG 1 RADIAL MESH')
      STOP
10    DO 20 J=2,JUP
      IF((Z(J).GE.H2).AND.(NZ.EQ.0)) NZ=J
      IF((NZ.EQ.J).AND.(ABS(Z(J)-H2).GT.ABS(Z(J-1)-H2))) NZ=J-1
      IF(NZ.NE.0) GO TO 25
20    CONTINUE
25    MINZ=NZ
      MAXZ=MINZ
30    MINR=0
      MAXR=0
      DO 35 I=2,IUP
      IF((R1(I).GE.RLO).AND.(MINR.EQ.0)) MINR=I
      IF((MINR.EQ.I).AND.(ABS(R1(I)-RLO).GT.ABS(R1(I-1)-RLO)))
&MINR=I-1
      IF(MINR.NE.0) GO TO 40
35    CONTINUE
      IF(RHI.GE.R1(IUP)) MAXR=IUP
      IF(MAXR.NE.0) GO TO 50
      MINRP1=MINR+1
40    DO 45 I=MINRP1,IUP
      IF((R1(I).GE.RHI).AND.(MAXR.EQ.0)) MAXR=I
      IF((MAXR.EQ.I).AND.(ABS(R1(I)-RHI).GT.ABS(R1(I-1)-RHI)))
&MAXR=I-1
      IF(MAXR.NE.0) GO TO 50
45    CONTINUE
50    WRITE(6,500) R1(MINR),R1(MAXR),Z(MINZ),Z(MAXZ)
500  FORMAT('//,8X,'THE BOUNDARIES FOR THE REGION TO BE SCANNED:',//,
      &6X,'MINIMUM R= ',E12.5,2X,'MAXIMUM R= ',E12.5,/,6X,'MINIMUM Z= ',
      &E12.5,2X,'MAXIMUM Z= ',E12.5)
      RETURN
      END
      SUBROUTINE TFORM1(EMU,ETA,XI,PHI,THETA,EMUP,EMUM,ETAP,ETAM)
      DIMENSION T(3,3)
      T(1,1)=COS(PHI)
      T(2,1)=-SIN(PHI)
      T(3,1)=0.0
      T(1,2)=SIN(PHI)*COS(THETA)
      T(2,2)=COS(PHI)*COS(THETA)
      T(3,2)=-SIN(THETA)
      T(1,3)=SIN(PHI)*SIN(THETA)
      T(2,3)=COS(PHI)*SIN(THETA)

```

```

T(3,3)=COS(THETA)
EMUP=T(1,1)*ETA+T(1,2)*XI+T(1,3)*EMU
ETAP=T(3,1)*ETA+T(3,2)*XI+T(3,3)*EMU
EMUM=T(1,1)*ETA-T(1,2)*XI+T(1,3)*EMU
ETAM=T(3,1)*ETA-T(3,2)*XI+T(3,3)*EMU
RETURN
END
SUBROUTINE TFORM2(EMU,ETA,XI,PHI,THETA,EMUP,EMUM,ETAP,ETAM)
GAMMA=THETA-PHI
T1=COS(GAMMA)
T2=-SIN(GAMMA)
EMUP=T1*EMU+T2*XI
ETAP=ETA
EMUM=T1*EMU-T2*XI
ETAM=ETA
RETURN      END
SUBROUTINE PCON(EMU,ETA,W,PL,PN,A,CP,MM,ISC,ISCT)
DIMENSION EMU(1),ETA(1),W(1),PL(1),PN(MM,ISC,1),A(1),CP(1)
C
C ---- PCON CALCULATES A(L,M,D) TERMS FOR INSCATTER INTEGRAL
C          USING VARYING ORDER RECURSION FORMULA
C          REFERENCE EQ. 8.5.1 - ABRAMOWITZ & STEGUN
C          THIS NEW VERSION CONTRIBUTED BY D. B. SIMPSON, 1975
C
IFM=2*ISCT+1
A(1)=1.0
DO 1010 I=2,IFM
1010 A(I)=FLOAT(I-1)*A(I-1)
DO 1030 M=1,MM
E1=0.0
IF(W(M).EQ.0.0) GO TO 1015
C      CHANGE PCON TO PREVENT NEGATIVE SQUARE ROOT ARGUMENT.
E1=SQRT(ABS(1.0-EMU(M)*EMU(M)-ETA(M)*ETA(M)))
1015 CP(M)=ATAN(E1/ETA(M))
IF(ETA(M).LT.0.0) CP(M)=CP(M)+3.1415927
PN(M,1,1)=1.0
PN(M,2,1)=EMU(M)
C      CHANGE PCON TO PREVENT NEGATIVE SQUARE ROOT ARGUMENT.
PN(M,2,2)=SQRT(ABS(1.0-EMU(M)*EMU(M)))
IF(ISCT.LE.1) GO TO 1030
ROOT=1.0/PN(M,2,2)
DO 1020 N=2,ISCT
E1=1.0-1.0/FLOAT(N)
E2=E1+1.0
PN(M,N+1,1)=E2*EMU(M)*PN(M,N,1)-E1*PN(M,N-1,1)
DO 1020 J=1,N
I1=N-J+1
I2=N+J-1
E1=I2
E2=I1
1020 PN(M,N+1,J+1)=ROOT*(E1*PN(M,N,J)-E2*EMU(M)*PN(M,N+1,J))
1030 CONTINUE

```

```

      I1=0
      DO 1040 M=1,MM
      LL=0
      DO 1040 L=2,ISC
      DO 1040 K=1,L
      LL=LL+1
      I1=I1+1
      PL(I1)=PN(M,L,K)
      IF(K.EQ.1) GO TO 1040
      IF((2.0*A(L-K+1)/A(L+K-1)).GT.0.0) GO TO 1035
      WRITE(6,222) L,K,(A(I),I=1,IFM)
222  FORMAT(/,7X,'L:',17X,'K:',17X,'A:',/,7X,E12.5,7X,E12.5,7X,
&E12.5,(/,45X,E12.5))
1035  E1=SQRT(2.0*A(L-K+1)/A(L+K-1))
      E2=K-1
      PL(I1)=E1*PL(I1)*COS(E2*CP(M))
1040  IF(ABS(PL(I1)).LE.1.0E-5) PL(I1)=0.0
      RETURN
      END
      SUBROUTINE WRBNDI(RNTFX,FLXIT,FLXJT,IGM,JM,IMA,MMA,NINTFX,
&NJNTFX,IG,NTIBO)
      DIMENSION RNTFX(1),          FLXIT(1),  FLXJT(1)
C
C - - - - WRITING OF BOUNDARY SOURCE OUTPUT FILE
C
      COMMON /COMIN/ TITL(12),TIMAR(10),MULT,LLKAR,NIN,NOU,NPU,
&NPR,JERR,ISIZE,NBYTE,NSEV,NCND,NCNDGO
      REAL*8 HNAME,HVERS,HNAMEX,TITL,TIMAR
      DIMENSION HNAME(17),IC(25)
      EQUIVALENCE (HVERS,IVERS)
      DATA HNAMEX /6HBNDRYS/, IVERS /0/,      IC /25*0/
1000  IF(IG.GT.1) GO TO 2000
      HNAME(1)=HNAMEX
      HNAME(2)=TITL(1)
      HNAME(3)=TITL(2)
      HNAME(4)=HVERS
      CALL SEQIO(2,NTIBO,HNAME,1+3*MULT,1)
      CALL CMOVX(TIMAR(4),HNAME,5*MULT)
      CALL CMOVX(TITL,HNAME(6),12*MULT)
      CALL SEQIO(2,NTIBO,HNAME,18*MULT,0)
      IC(1)=IGM
      IC(2)=JM
      IC(3)=IMA
      IC(4)=MMA
      IC(5)=NINTFX
      IC(6)=NJNTFX
      CALL SEQIO(2,NTIBO,IC,25,0)
      CALL SEQIO(2,NTIBO,RNTFX,NINTFX+NJNTFX,0)
C
2000  IF(NINTFX.EQ.0) GO TO 2100
      CALL SEQIO(2,NTIBO,FLXIT,MMA*JM*NINTFX,0)
      IF(LLKAR.LE.0) GO TO 2100
      CALL WOT3(FLXIT,0,JM,-MMA,4H M=,4H J=,NPR,

```

```

&-NINTFX,4HINT=,6H0FLXIT,6)
2100 IF(NJNTFX.EQ.0) GO TO 9900
      CALL SEQIO(2,NTIBO,FLXJT,MMA*IMA*NJNTFX,0)
      IF(LLKAR.LE.0) GO TO 9900
      CALL WOT3(FLXJT,0,IMA,-MMA,4H M=,4H I=,NPR,
&-NJNTFX,4HJNT=,6H0FLXJT, 6)
9900 RETURN
      END

```

```

//L      PROC UNT='3330-1',VLM=ZX1111,PAR=,DSNAM='X.WAR14636.PROG' L 0000
//LKED EXEC PGM=IEWL,REGION=150K,PARM='OVLY,MAP,LIST,LET,DCBS,&PAR' L 0010
//SYSPRINT DD SYSOUT=A,DCB=BLKSIZE=605 L 0020
//SYSLIB DD DISP=SHR,DSN=SYS2.FORTENHL
// DD DSN=SYS1.LOGLIB,DISP=SHR L 0040
// DD DSN=SYS1.CALLIB,DISP=SHR L 0050
//SYSLIN DD DSN=SYS1.LOADSET,DISP=(MOD,DELETE),UNIT=SYSDA, L 0080
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(TRK,0) L 0090
// DD DDNAME=SYSIN L 0100
//SYSLMOD DD UNIT=SYSDA,DISP=(,PASS),DSN=&&LM,SPACE=(TRK,(50,9,5),RLSE) *//
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD)),SPACE=(3072,(50,10)) L 0120
//PROG DD DISP=SHR,UNIT=&UNT,VOLUME=SER=&VLM,DSN=&DSNAM L 0130
// PEND L 0140
//LINK EXEC L *//
      INCLUDE PROG(QSAM)
      INCLUDE PROG(GIP) *//
      NAME COUPL(R) *//
//DOS EXEC DOS,REG=500K,PROG=COUPL,FT99=DUMMY
//GO.STEPLIB DD DISP=(OLD,DELETE),DSN=&&LM,SPACE=(TRK,(0,50,1)),
// UNIT=SYSDA
//GO.FT05F001 DD DDNAME=SYSIN
//GO.FT10F001 DD UNIT=3330V,DSN=TZA.WAR.SOL1,
// DISP=(MOD,KEEP),SPACE=(TRK,(20,20),RLSE),DCB=(RECFM=VBS,
// BLKSIZE=6144,LRECL=X,BUFNO=1),VOL=SER=VZX111
//GO.FT12F001 DD UNIT=3330V,DSN=TZA.WAR.BNDRY0,
// DISP=(NEW,KEEP),SPACE=(TRK,(20,20),RLSE),DCB=(RECFM=VBS,
// BLKSIZE=6144,LRECL=X,BUFNO=1),VOL=SER=VZX111
      COUPLING DOT CYLINDERS: VARFLM TO BNDRYS1$$ 1 10 12 91 96 10 10 40 10 14
2** 1.50E02 0.0 0.0
0T
3** 39I 0.0 200.0
4** 9I 0.0 50.0
5** 9I 0.0 50.0
6* 9I 0 3.1415927
/ S12 FULL SYM 1 2 13 SCS80010
7* SC120020
0 -24618- 5 0 -17408- 5 0 &17408- 5 0 -49237- 5 0 -46057- 5 0 -17408- 5SC120030
0 &17408- 5 0 &46057- 5 0 -65134- 5 0 -62765- 5 0 -46057- 5 0 -17408- 5SC120040
0 &17408- 5 0 &46057- 5 0 &62765- 5 0 -77850- 5 0 -75879- 5 0 -62765- 5SC120050
0 -46057- 5 0 -17408- 5 0 &17408- 5 0 &46057- 5 0 &62765- 5 0 &75879- 5SC120060
0 -88763- 5 0 -87039- 5 0 -75879- 5 0 -62765- 5 0 -46057- 5 0 -17408- 5SC120070
0 &17408- 5 0 &46057- 5 0 &62765- 5 0 &75879- 5 0 &87039- 5 0 -98473- 5SC120080
0 -96922- 5 0 -87039- 5 0 -75879- 5 0 -62765- 5 0 -46057- 5 0 -17408- 5SC120090
0 &17408- 5 0 &46057- 5 0 &62765- 5 0 &75879- 5 0 &87039- 5 0 &96922- 5SC120100

```

```

0 -24618- 5 0 -17408- 5 0 &17408- 5 0 -49237- 5 0 -46057- 5 0 -17408- 5SC120110
0 &17408- 5 0 &46057- 5 0 -65134- 5 0 -62765- 5 0 -46057- 5 0 -17408- 5SC120120
0 &17408- 5 0 &46057- 5 0 &62765- 5 0 -77850- 5 0 -75879- 5 0 -62765- 5SC120130
0 -46057- 5 0 -17408- 5 0 &17408- 5 0 &46057- 5 0 &62765- 5 0 &75879- 5SC120140
0 -88763- 5 0 -87039- 5 0 -75879- 5 0 -62765- 5 0 -46057- 5 0 -17408- 5SC120150
0 &17408- 5 0 &46057- 5 0 &62765- 5 0 &75879- 5 0 &87039- 5 0 -98473- 5SC120160
0 -96922- 5 0 -87039- 5 0 -75879- 5 0 -62765- 5 0 -46057- 5 0 -17408- 5SC120170
0 &17408- 5 0 &46057- 5 0 &62765- 5 0 &75879- 5 0 &87039- 5 0 &96922- 5SC120180
8* SC120190
3R-96922- 5 5R-87039- 5 7R-75879- 5 9R-62765- 511R-46057- 513R-17408- 5SC120200
3R&96922- 5 5R&87039- 5 7R&75879- 5 9R&62765- 511R&46057- 513R&17408- 5SC120210
T SC120370
=END
//

```

VITA

Jacob Frederick Mannes Schmidt was born in Oak Ridge, Tennessee on May 17, 1954. He attended elementary schools in that city and graduated from Oak Ridge High School in 1972. He recieved a Bachelor of Science degree in Mathematics from The University of Tennessee at Knoxville in March 1978. In the fall of 1978 he accepted a teaching assistantship at The University of Tennessee, and began study towards a Master's degree in Mathematics. In the fall of 1979 he accepted a research assistantship at The University of Tennessee which involved a project at the Oak Ridge National laboratory.

In February 1981 he accepted employment with Union Carbide at Oak Ridge National Laboratory as a computer analyst. He recieved a Master of Science degree in Mathematics in March 1982.

He is married to the former Schree Liafsha of Oak Ridge, Tennessee.