ORNL/TM-6529

The DOT-IV Two-Dimensional Discrete Ordinates Transport Code with Space-Dependent Mesh and Quadrature

W. A. Rhoades
D. B. Simpson
R. L. Childs
W. W. Engle, Jr.

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION - FOR THE DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, VA 22161
Price: Printed Copy $[illegible]; Microfiche $3.00

ORNL/TM-6529

Contract No. W-7405-eng-26

Engineering Physics Division

THE DOT-IV TWO-DIMENSIONAL DISCRETE ORDINATES TRANSPORT CODE WITH SPACE-DEPENDENT MESH AND QUADRATURE

W. A. Rhoades
D. B. Simpson
R. L. Childs*
W. W. Engle, Jr.

*Computer Sciences Division

JANUARY 1979

NOTE: Work supported by the DOE Office of Energy Technology, Division of Reactor Research and Technology; the Defense Nuclear Agency; and the U. S. Nuclear Regulatory Commission.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830

Operated by

UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

Table of Contents

Section 0

0.1 Abstract

DOT IV is designed to allow very large problems to be solved on a wide range of computers and memory arrangements. New flexibility in both space-mesh and directional-quadrature specification is allowed. For example, the radial mesh in an R-Z problem can vary with axial position. The directional quadrature can vary with both space and energy group. Several features improve performance on both deep penetration and criticality problems. The program has been checked and used extensively on several types of computers.

0.2 Foreword

In its present state of development, this preliminary report has been sufficient to guide use of the code at several laboratories. It is obviously lacking detail regarding theory and solution technique. It is It is our intention to publish a more complete document at a later time.

0.3 Acknowledgement: The authors wish to acknowledge the contributions of F. R. Mynatt, ORNL, who proposed and guided this project; Virginia Glidewell and Susan Engle for typing early drafts; and to Eddie W. Bryant, who prepared the manuscript for publication.

0.4 Limitations As of August 1978

All of the features have been insured operable at one time or another except two, which must not be used:

(1) Criticality searches, and

(2) P_L variable by group or material.

Problems using the discrete-ordinates method together with space-dependent rebalance and either periodic boundary conditions or internal

boundary sources will find convergence slow and erratic. This is due to limitations of the rebalance method used, and it may be corrected in later releases. Internal boundary source problems must also have XNF=0. The group source summary and the balance tables will not be correct.

Diffusion theory problems must not use internal or external boundary sources, variable mesh, or variable quadrature. A diffusion iteration cannot produce internal boundary source output or an "angular flux tape." The P_1 module is very limited, and its use is not recommended.

The special geometries, INGEOM$\geq$10, have not been completely checked and are not guaranteed.

Section 1. Program Abstract

1.1. Program: The DOT-IV Two-Dimensional Discrete Ordinates Transport Code.

1.2. Problem Solved: DOT IV determines the flux or fluence of particles throughout a two-dimensional geometric system due to sources either generated as a result of particle interaction with the medium, or incident upon the system from independent sources. The principal application is to the deep penetration transport of neutrons and photons. Criticality (k-type and search) problems can be solved. Numerous printed edits of the results are available, and results can be transferred to output files for subsequent analysis.

1.3. Method of Solution: The Boltzmann transport equation is solved using the method of discrete ordinates, diffusion theory, or a special "combined P_1" solution. In the discrete ordinates method, the primary mode of operation, balance equations are solved for the flow of particles moving in a set of discrete directions in each cell of a space mesh, and in each group of a multigroup energy mesh. Iterations are performed until all implicitness in the coupling of cells, directions, groups, and source regeneration has been resolved. Methods are available to accelerate convergence by space-dependent rebalance and by successive over-relaxation. Anisotropic cross sections can be expressed in a Legendre expansion of arbitrary order. Output data sets can be used to provide an accurate restart of a previous problem.

Special techniques are available to remove the effects of negative fluxes caused by the finite space and direction meshes, and of negative scattering due to truncation of the cross-section expansion. The space mesh can be described such that the number of first-dimension (I) intervals varies with the second dimension (J). The number of discrete directions can vary across the space mesh and with energy. The order of Legendre expansion can vary with cross-section set and with energy group.

Provision is made to treat sources resulting from the first collision of particles from a point source. In this case, flux due to uncollided particles is included in the output edits.

Direction sets can be biased, with more discrete direction segments per unit solid angle in some directions than in others.

1.4. Related Material:

Data Files

Microscopic cross-section input file
Independent source file (optional)
Flux guess input file (optional)
Flux result output file (optional)
Total source output file (optional)

Related Programs

GIP - prepares cross-section input from card or tape input
GRTUNCL - prepares first-collision source (IBM users only)
RTFLUM - edits flux files and converts to/from other file formats

1.5. Restrictions: External force fields or non linear effects cannot be treated. Flexible dimensioning is used throughout, so that no restrictions are imposed on individual problem parameters. Certain options, especially diffusion and P_1 theories, are not compatible with variable mesh and quadrature problems.

1.6. Computers: DOT IV is designed to be applicable to most sophisticated computers which support direct (random) access disk storage or the equivalent. It has special provisions for efficient use of a large, slow memory from which data are moved to fast memory in strings. Proper operation has been demonstrated on the IBM 360/75, 360/91, 360/195, 370/155, 370/168, and 3033 computers, and on the CDC 7600, CYBER 176, and STAR 100 computers.

1.7. Running Time: Running time is roughly proportional to:

Flux work units (FWU) = number of space mesh cells x number of directions x number of energy groups x number of iterations per group. Depending on the options chosen, a rate of 1.3 to 2.3 million FWU per minute on the IBM 360/195 is typical. Thus, a very large problem with 5,000 space cells, 48 directions, 50 energy groups, and 10 iterations per group would require roughly 1 to 1.5 hours of 360/195 CPU time.

1.8. Programming Languages: The program is operable with 100% FORTRAN language. The standard ANS STD. 3-1971[1] was followed. Machine-dependent features are localized and flagged with special comment cards. Roughly 10% of the FORTRAN has been translated to IBM assembler language for optional use, giving more than double the usual execution speed. The size of the routines which benefit by assembler language has been minimized to allow similar treatment on other machines.

1.9. Operating System: The IBM version uses OS Version 21. No special requirements are made of the operating system. Two types of overlay facility can be used if available.

1.10. Machine Requirements: Memory must be approximately 50,000 words for a small problem, expanding with problem size. External data storage must be provided for 8 scratch files, of which 4 must be direct (random) access. User-supplied input and output data files must be supplied on sequential-access devices, tapes, or the equivalent.

1.11. Authors: Currently responsible: W. A. Rhoades, Oak Ridge National Laboratory, P. O. Box X, Building 6025, Oak Ridge, TN 37830.

Other ORNL authors or major contributors: D. B. Simpson, F. R. Mynatt, W. W. Engle, Jr., and R. L. Childs.

Questions concerning the RSIC-distributed code package should be referred to the Radiation Shielding Information Center (RSIC), Oak Ridge National Laboratory, P. O. Box X, Oak Ridge, Tennessee 37830.

1.12. Reference:

W. A. Rhoades and F. R. Mynatt, "The DOT-III Two-Dimensional Discrete Ordinates Transport Code," ORNL-TM-4280 (September 1973).

F. R. Mynatt, F. J. Muckenthaler, and P. N. Stevens, "Development of a Two-Dimensional Discrete Ordinates Transport Theory for Radiation Shielding," CTC-INF-952 (August 1969).

1.13. Material Available: The code package is available from the Radiation Shielding Information Center (RSIC), Oak Ridge National Laboratory, P. O. Box X, Oak Ridge, Tennessee 37830. Approximately 43,000 logical records of information (FORTRAN source programs, optional IBM Assembler-Language routines, sample problem input and output) and a descriptive document (follows Standard ANSI N413-1974) are available. From one to three full reels of magnetic tape, dependent on a given computer installation environment, are required for transmission of the material.

1.14. Abstract Author: W. A. Rhoades, ORNL.

Section 2. Introduction

The primary objective of the DOT-IV project was to allow large shielding problems to be solved as a single unit. The demand for complexity and detail in analysis of shield systems had forced analysts to a "piecewise" approach. Several segments of a problem were solved separately, coupled at the boundaries in a non-iterative fashion. Changes in space mesh and directional quadrature were performed during the coupling steps. This system had many practical difficulties. Solving a problem in several segments routinely involved weeks of delay, since each segment tied up an entire computer for many hours, and each depended upon the results of the previous segment. Problem setup was complicated, as was assembling the fragmentary results into a single product.

The DOT IV code is designed to solve such problems as a unit. The long running time is made more acceptable by a reduction in memory requirement, allowing large problems to share the computer capacity with other jobs. Flexible storage capabilities, together with certain other novel features, make the code applicable to a wide range of problems, and operable on many dissimilar computer types.

DOT IV performs all of its computations in a small "working memory" sufficient to calculate fluxes along a single row of the space mesh. (Figure 2.1). This corresponds to "SCM" on a CDC-7600. The remaining memory is used as a "direct-access buffer" area to hold data for as many rows of the space mesh as will fit. A data-manager subroutine moves row data between buffer and working memory as needed. Thus, the logic of the problem-solution subroutines does not depend upon the organization of the buffer. Efficient input-output (I/O) routines move data corresponding to blocks of the space mesh between buffer and external storage devices. On CDC computers, this buffer concept makes effective use of the large, slow memory, "LCM." On IBM computers, it allows "chained execution" of the data transfers, so that a very large buffer can be filled while the CPU is at work on another job.

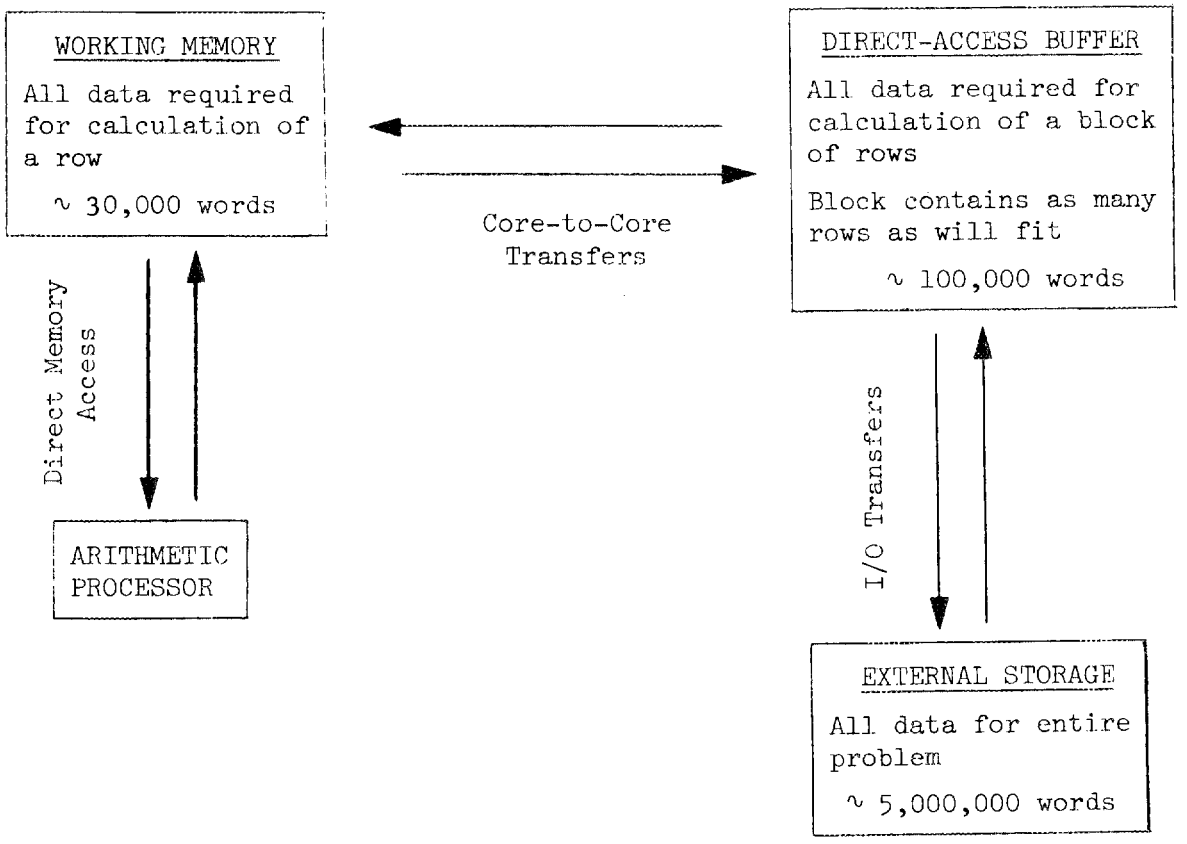


Fig. 2.1. DOT IV Split-Memory Storage Example

Section 3. Theoretical Considerations

3.1. Units DOT IV can run either in SI units or in traditional units:

	SI Units	Optional Traditional Units
Space Dimensions, R, Z, X :	Meters (m)	Centimeters (cm)
Time, t	Seconds (s)	
Rotation Fraction, θ :	Dimensionless	
Spherical Angle Fraction, W :	Dimensionless	
Atomic Densities :	$1/m^3$	$1/A^{o3}$ (Atoms per Cubic Angstrom)
Micro Cross Sections, σ :	m^2	barn (10^{-24} cm^2)
Macro Cross Sections, σ :	1/m	1/cm
Scalar fluxes, $\bar{N}$:	$1/m^2\cdot$	$1/cm^2\cdot s$
Directional fluxes, n :	$1/m^2\cdot s$	$1/cm^2\cdot s$
Boundary sources :	Same as directional fluxes	
Distributed sources :	$1/m^3\cdot s$ (per unit Z in XZ or Rθ geometries)	$1/cm^2\cdot s$

The directional weights sum to unity:

$$\Sigma W_m = 1$$

and directional fluxes are normalized such that, if the flux is isotropic, then the scalar and directional fluxes are equal in measure:

$$N = \Sigma W_m N_m$$

The $N_m/4\pi$ = directional flux in units of $1/m^2\cdot s\cdot Sr$; also, $\theta * 2\pi$ = rotational angle in units of radians (rad). If the time unit is dropped from the sources, then the fluxes become fluences, without the time unit.

3.2 Positive Solution of the Difference Equations

Development

The two-dimensional discrete-ordinates transport equation can be written: *

$$\mu\left(A_{i+1}N_{i+1} - A_iN_i\right) + \eta B\left(N_{j+1} - N_j\right) + \left(\gamma_{m+1}N_{m+1} - \gamma_mN_m\right) + \sigma NV = SV$$

- Known from problem data:

  A, B, γ: cell boundary geometric parameters
  σ: total cross section
  V: cell volume
  S: external source
  μ, η: direction cosines

  Known from previous boundary results:

  N_i, N_j, N_m: past boundary fluxes

  Unknown:

  N: average cell flux

  N_{i+1}, N_{j+1} N_{m+1}: new boundary fluxes

The simple step model can be obtained at once from this:

$$\text{Assume:} \quad N = N_{i+1} = N_{j+1} = N_{m+1}$$

$$\text{Then:} \quad N = \frac{SV + \mu A_iN_i + \eta BN_j + \gamma_mN_m}{\sigma V + \mu A_{i+1} + \eta B + \gamma_{m+1}}$$

*Only forms for $\mu>0$ and $\eta>0$ are shown. Other forms are very similar.

As shown in figure 3.1, the average flux is assumed equal to the final value at the cell boundary. This model is stable for all sizes of space cell and is often used as the "fixup" when other models generate negatives. It is not sufficiently accurate for general application, however.

The "diamond-difference" model can be regarded as equivalent to a linear variation of flux between boundaries with the average located at the midpoint (figure 3.2). Accordingly, it is sometimes called the "linear" model. To obtain it:

$$\text{Assume: } N = \frac{1}{2}\left(N_{i+1} + N_i\right) = \frac{1}{2}\left(N_{j+1} + N_j\right) = \frac{1}{2}\left(N_{m+1} + N_m\right)$$

$$\text{So that: } N_{i+1} = 2N - N_i; \quad N_{j+1} = 2N - N_j; \quad N_{m+1} = 2N - N_m$$

$$\text{Then: } N = \frac{SV + \mu\left(A_{i+1} + A_i\right)N_i + 2\eta B N_j + \left(\gamma_{m+1} + \gamma_m\right)N_m}{\sigma V + 2\left(\mu A_{i+1} + \eta B + \gamma_{m+1}\right)}$$

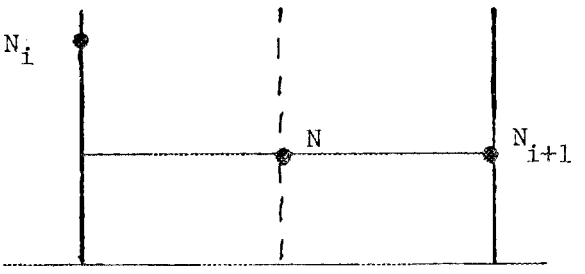


Figure 3.1. Step model

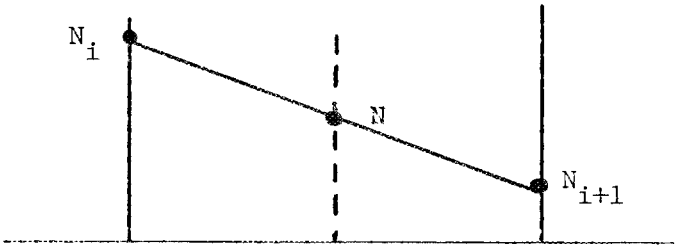


Figure 3.2. Diamond-difference (linear model)

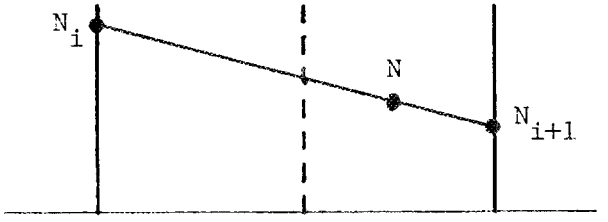


Figure 3.3. Weighted-difference model

The weighted-difference model uses three new parameters to interpolate between linear and step models (figure 3.3):

Assume:

$$a = \frac{N - N_i}{N_{i+1} - N_i} \; ; \quad b = \frac{N - N_j}{N_{j+1} - N_j} \; ; \quad c = \frac{N - N_m}{N_{m+1} - N_m}$$

So that:

$$N_{i+1} = \frac{1}{a}N - \left(\frac{1-a}{a}\right)N_i \; ; \quad N_{j+1} = \frac{1}{b}N - \left(\frac{1-b}{b}\right)N_j \; ; \quad N_{m+1} = \frac{1}{c}N - \left(\frac{1-c}{c}\right)N_m$$

Then:

$$N = \frac{SV + \mu\left(\frac{1-a}{a}A_{i+1} + A_i\right)N_i + \frac{1}{b}\eta B N_j + \left(\frac{1-c}{c}\gamma_{m+1} + \gamma_m\right)N_m}{\sigma V + \frac{1}{a}\mu A_{i+1} + \frac{1}{b}\eta B + \frac{1}{c}\gamma_{m+1}}$$

Defining: $M = 1/\left(\sigma V + \frac{1}{a}\mu A_{i+1} + \frac{1}{b}\eta B + \frac{1}{c}\gamma_{m+1}\right)$

Then:

$$N_{i+1} = \frac{M}{a}\left\{SV + \frac{1}{b}\eta B N_j + \left(\frac{1-c}{c}\gamma_{m+1} + \gamma_m\right)N_m + \left[\mu A_i - (1-a)\left(\sigma V + \frac{1}{b}\eta B + \frac{1}{c}\gamma_{m+1}\right)\right]N_i\right\}$$

$$N_{j+1} = \frac{M}{b}\left\{SV + \mu\left(\frac{1-a}{a}A_{i+1} + A_i\right)N_i + \left(\frac{1-c}{c}\gamma_{m+1} + \gamma_m\right)N_m + \left[\eta B - (1-b)\left(\sigma V + \frac{1}{a}\mu A_{i+1} + \frac{1}{c}\gamma_{m+1}\right)\right]N_j\right\}$$

$$N_{m+1} = \frac{M}{c}\left\{SV + \mu\left(\frac{1-a}{a}A_{i+1} + A_i\right)N_i + \frac{1}{b}\eta B + \left[\gamma_m - (1-c)\left(\sigma V + \frac{1}{a}\mu A_{i+1} + \frac{1}{b}\eta B\right)\right]N_m\right\}$$

Non-negativity can be assured by bounding a, b, and c between 1/2 (linear model) and 1 (step model); and, between those limits, to use:

$$1 - a = \frac{SV\theta_s + \left(\eta BN_j + \gamma_m N_m\right)\theta_n + \mu A_i N_i}{2\left(\frac{1}{2}\sigma V + \eta B + \gamma_{m+1}\right)N_i}$$

$$1 - b = \frac{SV\theta_s + \left(\mu A_i N_i + \gamma_m N_m\right)\theta_n + \eta BN_j}{2\left(\frac{1}{2}\sigma V + \mu A_{i+1} + \gamma_{m+1}\right)N_j}$$

$$1 - c = \frac{SV\theta_s + \left(\mu A_i N_i + \eta BN_j\right)\theta_n + \gamma_m N_m}{2\left(\frac{1}{2}\sigma V + \mu A_{i+1} + \eta B\right)N_m}$$

In the "super-weighted" mode, an arbitrary multiplier is used to avoid degeneracy and yet to use the accurate linear model when the source or flow from adjacent boundary is sufficient to guarantee positivity:

$$\theta_s = \theta_n = \theta.$$

Values of θ of 0.5 and 0.9 have been tested with satisfactory results. Apparently any value near 1, but not so near as to produce numerical degeneracy, is acceptable. An optimum value which gives best overall accuracy in the transition from linear to step may exist but has not been studied. A value of 0.9 is presently used in the code.

The DOT 3.5 code used:

$$\theta_s = 1 \; ; \; \theta_n = 0$$

and this is retained as the "weighted" mode in DOT IV. It provides more rapid convergence than other modes, and gives smooth spatial flux distribution in difficult deep penetration problems, where other modes may show spatial oscillation in the converged result. It is also better adapted to vectorization. Unfortunately, experience has shown that it is inaccurate in eigenvalue calculations.

3.3 Available Quadrature Sets

The direction variable $\vec{\Omega}$ in a radiation transport calculation is defined by its direction cosines with respect to the geometrical coordinate system. The possible orientations of the angular direction vector define a unit sphere in (μ,η,ξ) space. In a discrete ordinates calculation, this continuous direction space is represented by a discrete set of vectors known as the "discrete ordinates directional quadrature set:"

$$\hat{\mu}_m \hat{i} + \eta_m \hat{j} + \xi_m \hat{k}$$

with the constraint

$$\mu_m^2 + \eta_m^2 + \xi_m^2 = 1.0$$

The "weights" of the quadrature set, P_m, are proportional to the areas subtended by the solid angles associated with the specified directions. They correspond to fractions of the total surface area of the unit sphere and are generally positive with the normalization

$$\sum_m P_m = 1.0$$

Proper ordering of the directions is essential. For the DOT code and certain other codes, all directions with common η must be grouped together in order of increasing μ. Each such group must have, as its first member, an arbitrary boundary direction for which $\xi = 0$. Thus, for a given η:

$$\mu_0 = -\sqrt{1 - \eta^2}$$

These directions are assigned weight = 0. All negative η's must precede all positive η's. In a two-dimensional code, only 4 quadrants are considered, the others being redundant.

Full Symmetry

Fully symmetric quadrature sets are those exhibiting complete rotational symmetry; i.e., the discrete (μ_i, η_i, ξ_i) coordinates chosen to represent the direction vectors are required to be invariant under all 90 degree rotations about the μ, η, or ξ axis. Hence, each set of μ, η, ξ coordinates must be symmetric with respect to the origin; and, further, the set of projected points Θ_i on each axis must be the same. These conditions dictate that quadratures weights also be chosen in a symmetric fashion.

All Θ_i except Θ_1 are determined by the complete symmetry requirement and the fact that $\mu^2 + \eta^2 + \xi^2 = 1$. The selection of familiar Gaussian quadrature is not allowed, since choosing μ_i and η_i to be Legendre zeros, automatically sets

$$\xi_i = \pm\sqrt{1 - \mu_i^2 - \eta_i^2}$$

which is not a Gaussian set; and therefore, the Gaussian directions are not fully symmetric.

No Symmetry and Half Symmetry

Complete symmetry is required only in three-dimensional geometries. In lower dimensional geometries, a relaxation of symmetry requirements allows additional degrees of freedom.

A simple such relaxation is to keep the point and level arrangement of complete symmetry while allowing the points on each axis to be chosen from an independent set; the only requirement being that the points lie on the unit sphere; i.e.,

$$\mu^2 + \eta^2 + \xi^2 = 1$$

This corresponds to a "no symmetry" condition.

The "half symmetry" condition requires rotational symmetry about one axis only, so that $\mu_i^2 = \eta_i^2$, and $\xi_i = \pm\sqrt{1 - \mu_i^2 - \eta_i^2}$. Thus half symmetric quadrature sets are those which are invariant for 90° rotations only about the ξ axis. (Note the contrast to fully symmetric sets that exhibit rotational invariance about all three axes).

Unlike fully symmetric sets, the Θ's of a half symmetric set can be arbitrarily chosen. For an S_n half symmetric set, they are usually chosen to be the zeros of a N^{th} degree Legendre polynomial, and such was the case for the sets available with DOT IV.

Half symmetric quadrature sets available are the S4, S6, S8, and S10 sets and were generated by using the DOQDP[8] code. The N/2 positive roots of the N^{th} order Legendre polynomial (input descending order) were input as the values for Θ.

Biased Quadrature Sets

Biased quadrature sets are those quadrature sets which do not have an equal number of directions in the positive and negative domain of one of the variables; i.e., they are "biased" by having a larger number of directions in some portion of the unit sphere.

These sets are used when the neutron flow is highly anistropic in some preferred direction. The biased quadrature sets available are the 100, 166, and 210 direction downward biased sets and the 100, 166, and 210 direction upward biased sets.

100-Direction Biased Sets. The 100 direction sets contain 65 directions in the biased hemisphere and 35 directions in the unbiased hemisphere. The directions in the unbiased hemisphere were taken from the S10 half symmetric set. The directions in the biased hemisphere are also from the S10 half symmetric set; however, the first eta level, containing three points, has been replaced by 11 new levels, each containing three points. These 11 replacement levels were taken from a high order one-dimensional quadrature set.

166-Direction Biased Sets. The 166 direction sets contain 131 directions in the biased hemisphere and 35 directions in the unbiased hemisphere. The directions in the unbiased hemisphere were again taken from the S10 half symmetric set. The directions in the biased hemisphere are from a S10 half symmetric set, in which the first eta level, containing three points, has been replaced by 11 new levels, each containing nine points. These 11 new levels are again taken from a high order one-dimensional quadrature set.

210-Direction Biased Sets. The 210 direction sets contain 153 directions in the biased hemisphere and 57 directions in the unbiased hemisphere. They are found exactly like the 166 direction sets with one addition: the last eta level of both the biased and unbiased hemispheres, containing 11 points, is replaced by three new levels, each containing 11 points.

R-Theta Quadrature Sets

The sets available also contain the S2, S4, S6, S8, S10, S12, S14, and S16 R-θ quadrature sets.

An R-θ quadrature set is a set where the μ and $\S$ angles are specified instead of the μ and η angles; i.e., it corresponds to a slice through an infinite cylinder.

Since $\mu^2 + \eta^2 + \S^2 = 1$, given any two of the angles, the third one can be found. An R-θ quadrature set refers to the same directions as its counterpart R-Z quadrature set, but μ and $\S$ are used to specify the direction vector instead of μ and η. The direction ordering is not changed. The value of $\S$ is stored in the position normally occupied by η in data files and listings, and given the sign of the corresponding η.

The R-θ sets available with DOT IV were generated from the S2, S4, S6, S8, S10, S12, S14, and S16 fully symmetric R-Z quadrature sets. Note that the §'s for the zero-weight points are equal to ± 0.00001, rather than 0, to maintain the significance of the attached sign.

Section 4. Programmer's Information

4.1. Programming Language: The DOT-IV program is intended to be easily adaptable to various computers, and yet, to take advantage of high-performance structural features. The guidelines of ANS-STD.3-1971[1] are followed, in that the very simple FORTRAN language of ANSI X3.9-1966 is followed except where deviations:

(1) Provide important improvement in capability, and
(2) Can be localized and documented in some way.

The recommended procedures of the US DOE/DRRT Reactor Physics Branch Committee on Computer Code Coordination (CCCC), as reported in LA-5486-MS[2], have been heeded where applicable.

Where a few statements of machine-dependent or system-dependent coding are required, these are indicated by enclosing the statements in a pair of 3-character flags. The local adaptation programmer is then to choose the sets of statements which apply to his case and "comment out" the rest (hopefully by machine), e.g., if the code is set up for IBM operation, it might contain:

```
CIB
     ENTRY IBCDC(H,E,L,P)
CIB
CDC
C    ENTRY IBCDC
CDC
```

The CDC programmer should change this to:

```
CIB
C    ENTRY IBCDC(H,E,L,P)
CIB
CDC
     ENTRY IBCDC
CDC
```

Some rather complicated structural differences can be accommodated in this way. A list of flags used in DOT IV and a suggested list of choices for several applications are given in Table 4.1.

Table 4.1. Language Flags

		Application			
		A	B	C	D
CIB	IBM-Style Language	x			x
CDC	CDC-Style Language		x	x	
CSW	IBM-Style Word Size and System Features	x			
CLW	Non IBM-Style Word Size and System Features		x	x	x
CSG	CDC-Style Segment Loader is Used		x		
COV	CDC-Style Overlay Loader is Used			x	
CNO	CDC-Style Overlay Loader is Not Used	x	x		x
CSM	A Split Fast/Slow Memory is Used		x	x	
CNS	The Memory is Not Split	x			x

NOTE: A check means the feature is implemented on that type of application. Applications are:

A IBM
B CDC - Segment Loader Available
C CDC - No Segment Loader Available
D UNIVAC - No Overlay Used

No special demands are made of the system hardware or software environment other than the allocation of devices for input, output, and scratch data sets, and the support of a few system-oriented subroutines. These subroutines, specified or under review by CCCC, are to be supplied by the using installation. The corresponding routines are supplied with the code ONLY for the guidance of the programmer. Extensive comments in the routines, together with the discussions of LA-6941-MS[3], should enable most expert programmers to fulfill this requirement. The routines to which these comments apply are:

TIMER	Provides timing and job identification data
REED, RITE	Provide sequential access to data files
DRED, DRIT	Provide random access to data files
DOPC	Provides initiating, closing, and certain repositioning of data files
CRED, CRIT	Provide block transfer of data between fast and slow memory.

Subroutines called by the CCCC routines listed above may or may not be supplied for information only. Such subroutines include:

FBSAM, ITIME, IFTIME, ERRTRA, JOBNUM, and subroutines called by FBSAM.[7]

4.2 Overlay Arrangements

Two overlay arrangements are available with the code. The first, called the "segment structure," is applicable to IBM systems or to CDC systems which support the segmentation features of SCOPE 2.1. This structure is shown in Fig. 4.1.

The second arrangement is intended to allow operation on a CDC system which supports only the CALL OVERLAY statement. In this system, each program unit begins with a main program which has no arguments. The appropriate subprogram grouping is shown in Figure 4.2. This much simpler arrangement can also be used on IBM systems by choosing the second set of overlay cards, and this may have some advantage on certain systems.

Non-IBM, non-CDC systems can probably adapt to one of these two systems, but we have no applicable guidelines at this time.

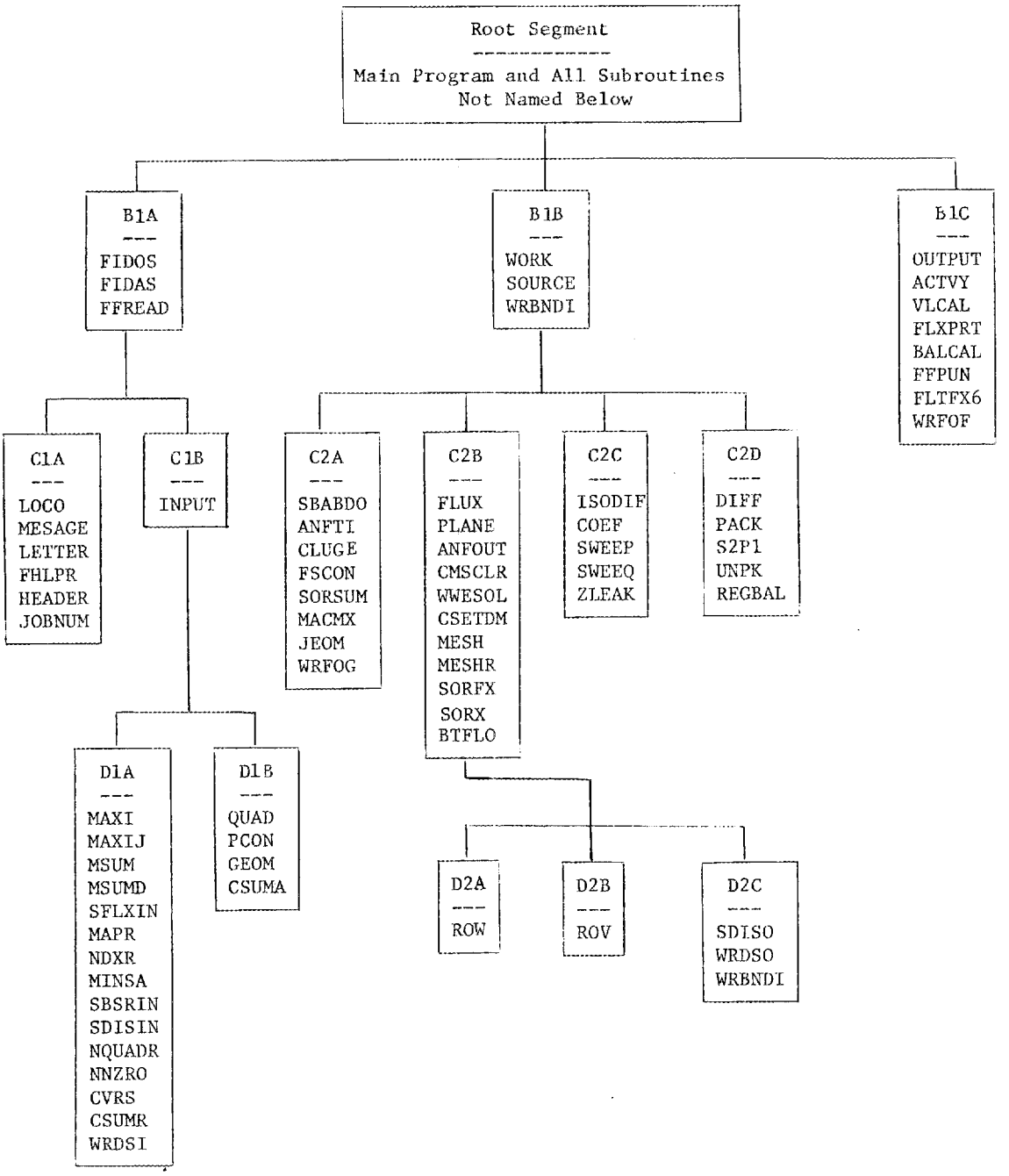


Figure 4.1. Segment Structure

<table>
<tr><td colspan="4">OVERLAY (0,0)

Contents of Root Segment in Previous Figure</td></tr>
<tr><td colspan="2">OVERLAY (1,0)

Contents of Segment

B1A</td><td>OVERLAY (2,0)

Contents of Segments

B1B
C2A
C2B
C2C
D2A
D2B
D2C
C2D</td><td>OVERLAY (3,0)

Contents of Segment

B1C</td></tr>
<tr><td>OVERLAY (1,1)

Contents of Segment

C1A</td><td>OVERLAY (1,2)

Contents of Segments

C1B
D1A
D1B</td><td></td><td></td></tr>
</table>

Figure 4.2. Overlay Structure

(Refer to previous figure for contents of segments)

4.3. Storage Allocation

Several labeled common blocks are used to contain individual data items and a few small arrays:

CMLOC	The locators specifying array origins in the container array. The first word is the BCD message "LOCL," while the second word is the length of the container array.
CMDOT	Input and code-generated problem parameters of general significance.
CMFLX	Parameters used primarily during the flux calculation.
IOREC	Data set status information.
COMIN	Standard input-output devices, error processing, titles, word length.
COMSAM	Status information for the random access routines. This is used only by the CCCC routines and may be removed during local adaptation.

All arrays of other than trivial length are assigned space in the container array at execution time; their lengths determined by the input data. In general, arrays are passed through an argument list before use. Dimensions not essential to execution are set to 1 to help in holding the number of arguments within limits.

On IBM systems, the length of the container array is adjusted at execution time to the space available (if the assembler-language ALOCAT routine is used). Space for system buffers to be used with the external data sets must be reserved by the parameter NBUF in the parameter input read by subroutine LOCO. Then ALOCAT acquires the necessary storage using a GETMAIN MACRO-instruction, and passes it as an argument to lower subprograms.

On other systems, the container array is located in blank common, and its size must be set at compilation time.

If the "split-memory" option is chosen, as in a CDC 7600 application, a large block of slow memory is assumed to be accessible through the CRED and CRIT routines. The length of this block is declared in subroutine ALOCAT. Following CCCC procedures,[3] data are transferred to and from slow memory only through the block move routines CRED and CRIT, providing top efficiency.

On some systems, slow memory is used as an extension of fast memory, accessed by ordinary FORTRAN statements. On such systems, the split-memory option should not be chosen. The split-memory option may reduce the maximum problem size in some cases.

4.4. Internal Data Management

The data storage is managed with only two major operating modes, a considerable simplification over some systems. The first mode, "problem stored," presumes that all flux moments and source moments, together with problem description arrays, will be stored in "working memory." Only incidental data arrays such as boundary sources and fluxes, cross sections, and scalar fluxes are kept on external devices, and "slow memory" is not used.

The second mode, "row stored," stores fluxes and source moments for only one row of space mesh cells in the working memory at a time. Flux and source moments for several rows or for the entire space mesh are stored in "user buffers." On split-memory systems, these are located in slow memory. On non-split systems, the user buffers are located in the container array beyond the "working memory" arrays. Data for each row are moved into working memory as required. The moving is managed by subroutine VARIO, and is accomplished by CRIT and CRED.

Best efficiency is obtained when an entire space mesh is contained in the user buffers, but as little as a few rows can be contained if required. Two storage criteria must be met. The first, and most stringent, is that the "working memory" arrays must fit the container array. The

second is that the buffer data must fit the user buffer space. The code automatically reblocks the space mesh to meet the latter criterion, and is usually successful in doing so. The volume of data transmission to the external devices is increased by storing less than the entire space mesh, but does not otherwise depend upon the number of blocks. The logical records become smaller as the number of blocks increases, and this may be important on some systems.

4.5. External Data Management

Data are stored externally on large-scale storage devices in units called "data sets" or "files." Each file is assigned a "logical unit number" less than 100, and is accessed by a "logical name" in the program into which is set the logical unit number. As a matter of convenience, user-supplied input files and result files may be spoken of as "tapes," while scratch files may be spoken of as "disks," although other equivalent storage can be used.

All data communication with tapes is sequential in nature, managed by the SEQIO routine, and accomplished by REED and RITE. The sequential routines supplied with the code should be directly applicable to all systems. Copious comment cards identify the function of these and other data-handling routines.

Scratch file data communication is channeled through VARIO and falls into three classes. The "sequential" files, numbered 81-90, can be processed adequately by SEQIO, REED, and RITE; and cards for this operation are included in VARIO. On IBM systems, a performance advantage is gained by routing such data through BLKIO, DOPC, DRED, and DRIT.

The "quasi-sequential" class could be processed adequately by SEQIO if it were possible to rewrite a record without destroying subsequent records. The boundary directional flux file is an example of such application. On IBM systems, this must be routed to BLKIO, and the

program supplied will do this. Such files are numbered 94-99. On other systems, it may be possible and advantageous to route quasi-sequential files to SEQIO by modifying VARIO.

The third class of scratch access, "blocked random" access, is also managed by VARIO. It is called to access all moments of flux or source for a row. When the first row of a block is read or the last block written, VARIO, if appropriate, transfers the block between the user buffer and disk, accomplishing this through BLKIO, DOPC, DRED, and DRIT. For each row, VARIO calculates the origin of the required data, and moves the data to the working-memory space required. A special flag indicates whether the rows are being read in ascending or descending order. Such files are numbered 91-93.

While the task of VARIO is complicated, the result is that the rest of the code knows nothing of the details of the blocked storage, and only two indexing schemes are required. The cost of the data moving between fast core and user buffer has proven to be almost unmeasurably small.

4.6. Programming Style

The programming style is greatly affected by the irregular mesh requirements. As an example, the simple task of finding the zone number for space cell (I,J) is normally:

```
DIMENSION IJZN(IM,JM)
K = IJZN(I,J)
```

If IM depends upon J, however, this is not adequate. Instead, we assemble "indexing pivot" arrays:

$$IBJ(J) = \sum_{I=0}^{J-1} IM(I); \quad IM(0) = 0$$

whereupon, in ANSI FORTRAN:

```
CALL SUBA(D(LIJZN),D(LIBJ),JM)
     .
     .
     .
SUBROUTINE SUBA(IJZN,IBJ,JM)
DIMENSION IJZN(1)
IJ=I+IBJ(J)
K=IJZN(IJ)
RETURN
END
```

Although cumbersome, this optimizes into code nearly as efficient as normal FORTRAN. Most of the data arrays are singly dimensioned because of the irregular mesh features.

A set of "scratch parameters," I1 through I20 and E1 through E9, are located in CMDOT for general use. These provide some space savings and help to distinguish parameters whose use is only over the span of a few statements from parameters of real significance to which mnemonic names are attached.

4.7. Special Language Requirements

It is essential to operation of the code that dimensions not needed, i.e., the length of singly dimensioned arrays and the last dimension of multiply dimensioned arrays, can be set to 1 if the arrays are only pseudonyms for space in the container array and if they are not used in READ or WRITE statements.

It is essential that a given area of the container can be referred to as either real or integer in various places, so long as the type is consistent with the data being stored, e.g., in the example above, the real array D contains integer data IJZN.

In a few areas, real-type statements such as A(I)=B(I) may be used to move integer data, or vice versa. But A(I)=-B(I) is never used in such a case.

In FIDAS, for example, the same array may be given two or more names in the argument list and either name may be used to update the information, according to type. Non-subscripted data items are never treated in this manner, however.

4.8. Code Structure

The code is arranged in four essentially independent units, as indicated in the overlay structure. These units, together with the "control subprogram" applicable, are:

(1) Parameter Input (LOCO) - input and edit of single parameters controlling problem size and logic path; printing of header messages.

(2) Array Input (INPUT) - input of primary dimension-setting arrays describing mesh, variation of directional quadrature over the mesh, and extent of moment expansion; secondary dimension setting arrays describing the direction sets and various coarse meshes; general arrays describing other problem conditions; and optional arrays for internal generation of flux and source data input files.

(3) Computation (WORK) - solution of the transport problem and gathering of certain data to be edited. Certain output data files are generated here if necessary.

(4) Output (OUTPUT) - provides final normalizing and editing of results. Writes output data files not handled by WORK.

Section 5. User's Information

5.1. Card Input Data

The card input data for a problem consist of a title card followed by a variable number of blocks of data separated by a "T" delimiter. Within each block, a variable number of data arrays are specified, keyed according to an array number, and identified as to real or integer type.

A separator card must be placed between successive problem decks. If it contains the word "DIAG" in the first four columns, a non-fatal diagnostic will be given by ERRO. If the word "DUMP" appears, a full fatal dump is given. Otherwise, the next problem begins immediately. No data are retained in memory between problems. No further execution is possible after a problem encounters a fatal error.

5.2. Space Mesh Grouping

Several groupings of space mesh cells are used. The most familiar is the assignment of each space mesh cell to a "material zone." Cross-section sets and buckling are then specified by material zone.

One or more zones may be assigned to a "region" for balance table and activity output.

An arbitrary "coarse mesh" is specified for the spatial rebalance operation. A minimum of five storage locations are required for each space mesh cell and a ratio of three or more fine-mesh intervals per coarse mesh provides important space savings. The coarse mesh designation can be relatively arbitrary, since only the convergence speed is affected. Experience on a given class of problems is the best guide in this matter.

Another arbitrary mesh, the "super mesh" is similarly defined for search specifications and for specifying various directional quadrature sets. "Super groups" can also be defined, so that quadrature can depend upon both space and energy.

5.3. Irregular Dimensioning Specifications

The space mesh can be specified such that the I mesh depends upon J. To accomplish this, several "I-sets" are specified, with different numbers of intervals and different interval boundaries. The ISET(J) array then relates the proper set of boundaries to the J-level. Prudent use of this feature can save computation time and storage by concentrating work in areas requiring a fine mesh, but too-frequent changes can waste time.

The directional quadratures, "M-sets," are specified by super-zone and super-group, as discussed earlier. This allows the user to, for example, specify a highly biased quadrature in the area of a streaming crack without wasting such detail elsewhere in the problem. A certain amount of work is required to perform the translation between sets, however, and so the number of switches in the mesh should be minimal.

In addition, the order of moment expansion for each cross section is specified by material, and the order of flux expansion is specified by group. The cross-section expansion is largely a matter of user convenience when using data of mixed expansion or in testing the effect of various expansions, but the flux expansion specification can save I/O charges and storage. The expansion required at intermediate and lower energies is generally lower than the maximum. The cost of these two features is trivial.

There are certain restrictions on the mesh sets. The largest I-set and the largest M-set must fall within the input limits |IM| and |MM|. Likewise, the maximum cross-section expansion and the maximum flux expansion order must each be no more than |ISCTM|. Outer boundaries of I-sets must match, and coarse-mesh and super-mesh boundaries must be contained in all I-sets.

"Standard" I-sets and M-sets are specified in which the boundary sources and fluxes are kept. The use of a small "Standard M-set" with a problem using a very large M-set in the interior could eventually save much space for certain problems. For now, a restrictive set of rules prevents this until the restrictions are removed. The rules are:

(1) Standard set must have as many directions up and as many down as any set.
(2) Standard set must have |MM| directions.
(3) Standard set must have as many levels as any set.
(4) Standard set must be set #1 (first set) for maximum computing speed.

5.4. Card Input Format

With the exception of the title card, all data are read in the FIDO system also used in ANISN,[4] DOT III,[5] and other codes.

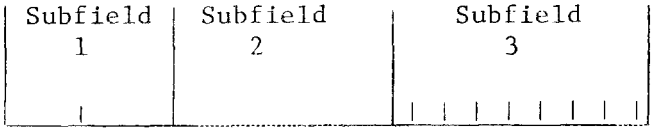
The FIDO input method is especially devised to allow the entering or modifying of large data arrays with minimum effort. Special advantage is taken of patterns of repetition or symmetry wherever possible. The FIDO system was patterned after the input method used with the FLOCO coding system at Los Alamos, and was first applied by Atomics International to the DTF-II[6] code. Since that time, numerous features requested by users have been added, a free-field option has been developed, and the application of FIDO has spread to innumerable codes.

The data are entered in units called "arrays." An array comprises a group of contiguous storage locations which are to be filled with data at one time. These arrays usually correspond on a one-to-one basis with FORTRAN arrays used in the program. A group of one or more arrays read with a single call to the FIDO package forms a "block," and a special delimiter is required to signify the end of each block. Arrays within a block may be read in any order with respect to each other, but an array belonging to one block must not be shifted to another. The same array can be entered repeatedly within the same block. For example, an array

could be filled with "0" using a special option, and then a few scattered locations could be changed by reading in a new set of data for that array. If no entries to the arrays in a block are required but the condition requiring the block is met, the delimiter alone satisfies the input requirement.

Three major types of input are available: fixed-field input, free-field input, and user-field input.

Fixed Field Input - Each card is divided into six 12-column data fields, each of which is divided into three subfields. The following sketch illustrates a typical data field. The three subfields always comprise 2, 1, and 9 columns, respectively.



To begin the first array of a block, an array originator field is placed in any field on a card:

Subfield 1: An integer array identifier < 100 specifying the data array to read.
Subfield 2: An array-type indicator -
"$" if the array is integer data
"*" if the array is real data
Subfield 3: Blank

Data are then placed in successive fields until the required number of entries has been accounted for. A sample data sheet shown below illustrates this input.

In entering data, it is convenient to think of an "index" or "pointer" which is under control of the user, and which specifies the position in the array into which the next data entry is to go. The pointer is always

positioned at array location #1* by entering the array originator field. The pointer subsequently moves according to the data operator chosen. Blank fields are a special case, in that they do not cause any data modification and do not move the pointer.

A data field has the following form:

Subfield 1: The data numerator, an integer < 100. We refer to this entry as N_1 in the following discussion.

Subfield 2: One of the special data operators listed below.

Subfield 3: A nine-character data entry, to be read in F9.0 format. It will be converted to an integer if the array is a "$" array or if a special array operator such as "Q" is being used. Note that an exponent is permissible but not required. If no decimal is supplied, it is assumed to be immediately to the left of the exponent, if any; and otherwise to the right of the last column. This entry is referred to as N_3 in the following discussion.

A list of data operators and their effect on the array being input follows:

"Blank" indicates a single entry of data. The data entry in the third subfield is entered in the location indicated by the pointer, and the pointer is advanced by one. However, an entirely blank field is ignored.

"+" or "-" indicates exponentiation. The data entry in the third field is entered and multiplied by $10^{\pm N_1}$, where N_1 is the data numerator in the first subfield, given the sign indicated by the data operator itself. The pointer is advanced by one. In cases where an exponent is needed, this option allows the entering of more significant figures than the blank option.

"&" has the same effect as "+" on IBM systems.

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

"R" indicates that the data entry is to be repeated N_1 times. The pointer is advanced by N_1.

"I" indicates linear interpolation. The data numerator, N_1, indicates the number of interpolated points to be supplied. The data entry in the third subfield is entered, followed by N_1 interpolated entries equally spaced between that value and the data entry found in the third subfield of the next non-blank field. The pointer is advanced by $N_1 + 1$. The field following an "I" field is then processed normally, according to its own data operator. The "I" entry is especially valuable for specifying a spatial mesh. In "$" arrays, interpolated values will be rounded to the nearest integer.

"L" indicates logarithmic interpolation. The effect is the same as that of "I" except that the resulting data are evenly separated in log-space. This is especially convenient for specifying an energy mesh.

"Q" is used to repeat sequences of numbers. The length of the sequence is given by the third subfield, N_3. The sequence of N_3 entries is to be repeated N_1 times. The pointer is advanced by N_1*N_3. If either N_1 or N_3 is 0, then a sequence of $N_1 + N_3$ is repeated one time only, and the pointer is advanced by $N_1 + N_3$. This feature is especially valuable for geometry specification.

"G" has the same effect as Q, except that the sign of the sequence is changed each time it is entered.

The "N" option has the same effect as "Q", except that the order of the sequence is reversed each time it is entered. This is valuable for the type of symmetry possessed by quadrature coefficients.

"M" has the same effect as "N" except that the sign of each entry in the sequence is reversed each time the sequence is entered. For example, the entries:

1 2 3 2M2

would be equivalent to:

1 2 3 -3 -2 2 3

This option is also useful in entering quadrature coefficients.

"Z" causes $N_1 + N_3$ locations to be set to 0. The pointer is advanced by $N_1 + N_3$.

"C" causes the position of the last array item entered to be printed. This is the position of the pointer, less 1. The pointer is not moved.

"O" causes the print trigger to be turned on. The trigger is originally off. When the trigger is on, each card image is listed as it is read.

"P" causes the print trigger to be turned off.

"S" indicates that the pointer is to skip N_1 positions leaving those array positions unchanged. If the third subfield is non-blank, that data entry is entered following the skip, and the pointer is advanced by $N_1 + 1$.

"A" moves the pointer to the position N_3, specified in the third subfield.

"F" fills the remainder of the array with the datum entered in the third subfield.

"E" skips over the remainder of the array. The array length criterion is always satisfied by an "E", no matter how many entries have been specified. No more entries to an array may be given following an "E", except that data entry may be restarted with an "A".

The reading of data to an array is terminated when a new array origin field is supplied, or when the block is terminated. If an incorrect number of positions has been filled, an error edit is given, and a flag is set which will later abort execution of the problem. FIDO then continues with the next array if an array origin was read. Otherwise, it returns control to the calling program.

A block termination consists of a field having "T" in the second subfield. All entries following "T" on a card are ignored, and control is returned from FIDO to the calling program.

Comment cards can be entered within a block by placing a slash (/) in column 1. Then columns 2-80 will be listed, with column 2 being used for printer carriage control. Such cards have no effect on the data array or pointer.

Free-field Input - With free-field input, data are written without fixed restrictions as to field and subfield size and positioning on the card. The options used with fixed-field input are available, although some are slightly restricted in form. In general, fewer data cards are required for a problem, the interpreting print is easier to read, a card listing is more intelligible, the cards are easier to keypunch, and certain common keypunch errors are tolerated without affecting the problem. Data arrays using fixed- and free-field input can be intermingled at will within a given block.

The concept of three subfields per field is still applicable to free-field input, but if no entry for a field is required, no space for it need be left. Only columns 1-72 may be used, as with fixed-field input.

The array originator field can begin in any position. The array identifiers and type indicators are used as in fixed-field input. The type indicator is entered twice, to designate free-field input (i.e., "$$" or "**"). The blank third subfield required in fixed-field input is not required. For example: 31** indicates that array 31, a real-data array, will follow in free-field format.

Data fields may follow the array origin field immediately. The data field entries are identical to the fixed-field entries with the following restrictions:

(1) Any number of blanks may separate fields, but at least one blank must follow a third subfield entry if one is used.
(2) If both first and second subfield entries are used, no blanks may separate them, i.e., 24S, but not 24 S.
(3) Numbers written with exponents must not have imbedded blanks, i.e., 1.0E+4, 1.0E4, 1.0+4, or even 1+4, but not 1.0 E4.
(4) In third-subfield data entries, only 9 digits, including the decimal but not including the exponent field, can be used, i.e., 123456.89E07, but not 123456.789E07.
(5) The Z entry must be of the form: 738Z, not Z738 or 738 Z.
(6) The + or - data operators are not needed and are not available.
(7) The Q, N, and M entries are restricted: 3Q4, 1N4, or M4, but not 4Q, 4N, or 4M. G is similarly restricted.
(8) A field must not span two cards.
(9) All items on a card entered after a slash in any column except the first are ignored.

User-Field Input - If the user follows the array identifier in the array originator field with the character "U" or "V", the input format is to be specified by the user. If "U" is specified, the FORTRAN format to be used must be supplied in columns 1-72 of the next card. The format must be enclosed by the usual parentheses. Then the data for the entire array must follow on successive cards. The rules of ordinary FORTRAN input as to exponents, blanks, etc., apply. If the array data do not fill the last card, the remainder must be left blank.

"V" has the same effect as "U" except that the format read in the last preceding "U" array is used.

Name General Example of Fido Input Charge Date Page

	IDENTIFICATION	REMARKS (DO NOT PUNCH)
1$		Begin the 1$ array, fixed-field, integral
1		Enter 1.
F2		Fill array with 2.
2*		Begin the 2* array, fixed-field, real.
1.234		Enter 1.234.
12.34 -1	10	" "
5- 1234 +02		" "
3- 1234		" "
7		" 7.0
		A blank field is always ignored.
T		Terminate this block.
	20	No entries may follow T on a card.
3*		Begin 3* array, fixed-field real.
9I 0		Enter 0,1,2,3,4,5,6,7,8,9,10,10,10.
3R 10		as real numbers.
3*,* 10S10		Repeat 3* in free-field, skip
11 12		to 11th entry, correct sequence to
	30	----9,10,11,12.
4,* 21 1.4.0		Begin 4* array, free-field, real.
2Q4		Enter 1,2,3,4, 1,2,3,4, 1,2,3,4.
E		End reading this array; remainder of array unchanged.
T		Terminate this block.
	40	

R - REPEAT I - INTERPOLATE S - SKIP T - TERMINATE

5.5. 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 Flux 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)

IADJ 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 NTSIG (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-θ, 180°-360° Triangular, 60° Triangular, 90° Triangular, 120° Triangular (Also See Data Note 1)

IBL Left Boundary Condition
IBR Right Boundary Condition
IBB Bottom Boundary Condition
(20) IBT Top Boundary Condition

0=Void 4=Fixed Boundary Source
1=Reflected 5=Albedo
2=Periodic
3=Cylindrical

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/DB**2 Search/ Concentration Search/Dimension Search

IACC 0/1/2 = Rebalance Method - Groupwise/Source Correction Method*/Space-Dependent Scalar 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 NINTFX I-Boundaries Written on NTIBO (May be 0)

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

IRED Number of Regions For Activity and Balance Table Output

IPDB2 0/1/2/3=No Effect/Punch 1 DB2 Value/IGM Values
IGM*IRED 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 χ Normalized/Fission With
Input χ/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 1/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")

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

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 71$ [1 if IM>0]
MSM = # non-zero entries in 72$ [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 MM>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(M,MSET) (#=MMSMSM)
Directional weights

82* EMU(M,MSET) (#=MMSMSM)
η, cosine of angle with 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 IACT≠0]
Region number by zone (Default = 1 region per zone)

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, χ, by group (#=IGM)

2* ZIN(J) Y, Z, or θ space-mesh boundaries (#=JM)

3* DNIJ(I,J) density factor (#=IMSJM) [IDFAC>0]
(Default=1.0)

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) [IFDB2Z>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) mixture 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 (#=NISZN)
[KTYPE=4]

20* ABDOL(IG,J) left boundary albedo (#=IGM*JM) [IBL=51]

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 IEPSBZ>0]

25$ ICMAT(IAC) material to be used in activity calculations (#=|IACT|) (See Data Note 4)

26$ ICPOS(IAC) cross section table position for activity (#=|IACT|)

27* ACMUL(IAC) activity multiplier (#=|IACT|)

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 direction. If $\mu>0$, the source applies to the left boundary; if $\mu<0$, to the right. Likewise, if $\eta>0$, a J-boundary source applies to the bottom; if $\eta<0$, 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 IADJ>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≥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 INGEOM=30, emerging flux will be calculated for a source entering each group JDIRF≤IGI≤JDIRL and leaving all groups IGI≤IGØ≤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≤INGEOM≤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:

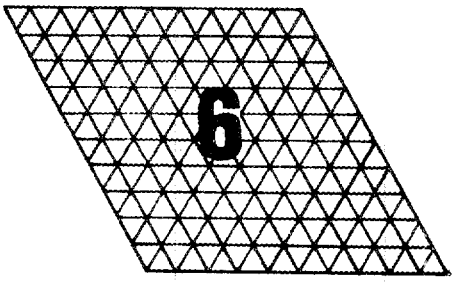
INGEOM	TRIANGULAR GEOMETRY OPTION	VALUE OF IM
3	180°-360° Symmetry	User's option
4	60° Symmetry	2*JM-1
5 (not oper)	90° Symmetry	2*JM+1
6	120° Symmetry	2*JM

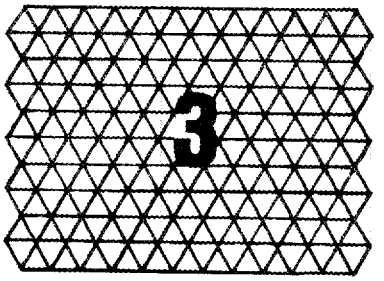
The resulting geometry is illustrated in Figure 5.1.

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 but 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 jth 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.





INGEOM	SYMMETRY
3	180-360
4	60
6	120

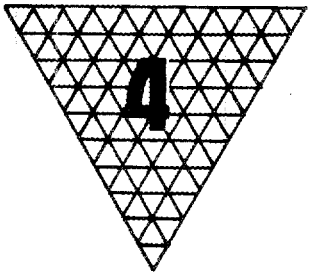


Figure 5.1

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 IGTYPE>0, then the first IGTYPE 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 IGTYPE 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 $0 \leq ORF \leq 1$ (RV = 0.5)

If combined P_1 theory is chosen, the code will automatically revert to row-stored 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 IZMT 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 ISCTM>0, then 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 these 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:

MIXT	NUCL	DENS
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 principal 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.

I	MIXT	NUCL	DENS	MATL
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, 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:

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

If σ_g^T is replaced by a "transport cross section" in reactor core problems, $\sigma_{g\to g}$ 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:

	ICMAT	ICPOS	ACMUL
	r	s	t

$$A_{i,j} = t \sum_{j} \sum_{i} d_{i,j} \; C_{i,j,r,s} \sum_{g} \phi_{g,j,i} \; \sigma_{g,r,s}$$

$$Bz = \sum_{i,j} X_{i,j,z} \; A_{i,j} \; V_{i,j}$$

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 of r to the macro according to the mixing table. Suppose IACT=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:

I	ICMAT	ICPOS	ACMUL
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 $|s| >$ IHP, then the value of 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.

5.6. CPU Time Usage

A reasonably accurate time limit should be calculated and entered as TMAX for all problems that run more than a few minutes. The maximum time given to the operator or operating system should be a few minutes larger than TMAX. This allows DOT to complete its I/O and data summaries if the time is exceeded during the iteration phase. The CPU time required, C, can be calculated by the following formula:

$$C = A + \left[\frac{FW}{FR}\right] * \frac{150}{SF}$$

$$FW = \sum_{P} \sum_{G} \sum_{F} \sum_{J} \sum_{I} MM$$

where

FW = flux work

MM = number of directions

I,J = space-mesh indices

G = energy group

F = inner (flux) iteration index

P = outer (source) iteration index

FR = flux rate (depends upon problem and code configuration)

SF = the speed factor (machine dependent)

A = overhead (I/O, problem setup, and summary activities)

The value of A is typically 0 to 2 minutes on the IBM 360/91. The table of suggested speed factors, drawn from experience with various codes, is:

Speed Factor

(Based on IBM 709 = 1)

IBM 360/195	300
IBM 370/168	150
IBM 360/91	150
IBM 360/75	30
IBM 360/65	20
IBM 7094	10
IBM 7090	5
CDC CYBER 176	600
CDC 6600	30
UNIVAC 1108	30

The value of FR depends upon the performance of the flux routine being used, and, to a lesser extent, upon the class of problem solved. On the IBM machines with the assembler-language routines installed, FR is about 1.0 million fluxes per minute for large P_3 problems using the linear difference model. The weighted model runs roughly 30% slower, and a P_o calculation runs as much as 40% faster. With optimized FORTRAN routines, FR would drop to something less than 0.5 million on either IBM or CDC machines. The combined P_1 routines have speed equivalent to MM at about 4. Assembler language routines are not available for CDC use. Expert opinions offered to us indicate that the CDC performance would not benefit greatly by assembler language.

5.7. Scratch Data Sets

The following lists the scratch data sets, the length of each record, and the number of records:

Lun	Logical Unit Symbol	Description	Record Length ℓ rec	No of Records n rec	Required
81	NDFIJ	Scalar Flux Scratch	IMA*JM	IGM	Always
82	NDSIG	Cross-Section Scratch	IHM*MTM	IGM	Always
83	NDBTL	Balance Table Scratch	NREG*10+IZM+8	IGM	IRED>0
84	NDBSI	Boundary Source Scratch	(IM+JM)*MMA	IGM	NTBSI>0
91	NDFLX	Flux Scratch	IM*LMX*JBLK1	NJBLK*IGM	NJBLK>0
92	NDSOR	Total Source Scratch	IM*LM*JBLK1	NJBLK	NJBLK>0
93	NDSIN	In-Source Scratch	IMA*LM*JBLK1	NJBLK	NJBLK>0
94	NDBFX	Boundary Flux Scratch	(IMA+JM)*MMA	IGM	Always

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

JBLK1 = #J levels per J block $1 \le JBLK1 < JM$
NJBLK = #J blocks $MINBLK \le NJBLK \le MAXBLK$ (NJBLK=0 if fluxes are stored in working memory)
LM = maximum number of moments (including 0th) in spherical harmonic expansions. $LM = 1 + |ISCT|*(|ISCT|+3)/2$
LMX = as LM, but variable by group. Thus each record may be of different length.
NREG = number of edit regions
IZM = number of material zones

The number of space blocks is controlled by input data. If MINBLK=0, the code will attempt to hold all flux and source data in fast memory. Failing in this, it will attempt to allocate space for the entire mesh to be held as 1 block in the user-buffer area (if MAXBLK≥1). Failing in this, the space mesh is broken into as many blocks as required to a limit of MAXBLK.

5.8. System Buffer Space

If the IBM assembler language ALOCAT subroutine is used, the space for system buffers and system control blocks used by all data sets except the card input and printed output must be allocated by specifying NBUF in the input parameter data. For each user-supplied data set, the space required in K-byte units (1 K-byte = 1024 bytes) is:

$$\text{K-bytes required} = 2+2*\left[\frac{\text{BLKSIZE*BUFNO}}{2048}\right]$$

where [] indicates truncation, and BLKSIZE and BUFNO are as specified in the DCB field of the JCL card. If BUFNO is not specified, assume 2.

For each scratch data set, where ntrk is the number of disk tracks required for a logical record:

$$\text{K-bytes required} = 2+2*\left[\frac{205*\text{ntrk}}{2048}\right] \text{ (approx)}$$

Non-IBM users should set NBUF=0.

5.9. Total Clock Time

The actual clock time, T, required for a solution can be established by:

$$T = C + W + IO$$

where:

W = wait time caused by other jobs running concurrently

IO = time required for disk-file access

IO is usually negligible in the "problem-stored" mode. In other modes, it can be estimated by:

$$IO = \sum_{P}\sum_{G}\sum_{GF}\left[LM(GF) + D1*II*LM(G)\sum_{J}\sum_{I}\frac{1}{DR}\right]$$

LM = number of moments of flux and source data, including the scalar

DR = disk data flow rate

D1 = 0 if data for the entire space mesh are stored in one block

= 11 otherwise

Obviously DR and W depend upon the system. In a typical very large problem using 1/2 of the memory of the Oak Ridge IBM 360/195, the space mesh was broken into 4 blocks, and the job ran in competition with other jobs. Approximately 20% of the time was consumed by W; 40% by C, and 40% in actual data transmission. If the entire memory is used, and the problem is run in group-stored mode, it is not unusual for C to approach 80% of T. DR is about 6×10^6 words per minute, but 10^7 should be available with improved software.

5.10. Memory Requirement

The exact data storage requirement can be found by examining the subroutine INPUT. It is a complex function of many input parameters, due to careful attention to space-saving techniques. A suitable estimate can

be obtained by examining only the major arrays. In the problem-stored mode, the total space requirement is approximately:

$$MM*(JM+IMA) + 2*IMSJM + 6*ICM*JCM + LM*IMSJM*(2+IGM)$$

In row-stored mode, the fast memory requirement is reduced to, approximately,

$$MM*(JM+IMA) + 2*IMSJM + 6*ICM*JCM$$

and the user-buffer space is:

$$LM*IMA*3*JBLK1$$

If fission density is calculated (NOFIS<2), The user-buffer space is increased by 2*IMSJM. Also, special options such as diffusion theory, variable mesh or quadrature, or boundary sources can increase this requirement. Users who are specifying diffusion theory for the entire problem should minimize storage requirements by specifying a coarse mesh such that ICM=JCM=1.

The total memory requirement includes, of course, the data storage requirement + system buffer space (if any) + space for program instructions.

5.11. Input and Output Data Files

All of the user input and output data files are expressed in formats designed according to recommendations of DOE's Committee on Computer Code Coordination (CCCC).[2] The specifications of these files are contained on the following pages. Two exceptions, retained for backward compatibility, are an optional capability to process a "GIP" cross section input file and to produce a "DOT Angular Flux Tape" as output.

The GIP file is defined exactly as the records labeled "CROSS SECTION DATA" in the ORDOSW file, and has no other record types. The "DOT Angular Flux Tape" is defined in the DOT III memo, ORNL-TM-4280[5] and is simulated by DOT IV as accurately as possible.

The correspondence of file type to use in the code is:

SYMBOL	FILE TYPE	USE
NTFLX	VARFLM	Flux and moment input
NTFØG	VARFLM	Flux and moment output
NTSIG	ORDOSW or GIP	Cross section input
NTBSI	BNDRYS	External boundary source input
NTDSI	VARSOR	Distributed source moment input
NTFCI	VARSOR	First collision source input
NTIBI	BNDRYS	Internal boundary source input
NTIBØ	BNDRYS	Internal boundary flux output
NTNPR	Formatted output	
NTDIR	DOT III "Angular Flux Tape"	Directional flux output
NTDSØ	VARSOR	Directional flux moment output

```
C                                                                       -VSOR0010
                                                                         VSOR0020
C***********************************************************************VSOR0030
C                        REVISED 10 NOV 76                              -VSOR0040
C                                                                       -VSOR0050
CF          VARSOR                                                      -VSOR0060
CE          VARIABLE MESH SOURCE MOMENT DATA                            -VSOR0070
C                                                                       -VSOR0080
C***********************************************************************VSOR0090
                                                                         VSOR0100
CD                       ORDER OF GROUPS IS BY DECREASING ENERGY         VSOR0110
CD                       I IS THE FIRST-DIMENSION INDEX                  VSOR0120
CD                       J IS THE SECOND-DIMENSION INDEX                 VSOR0130
CD                       JM=1 FOR 1-DIMENSIONAL GEOMETRY                 VSOR0140
CD                       IF ISOP.GT.0, SOURCE IS FIRST-COLLISION TYPE    VSOR0150
                                                                         VSOR0160
C-----------------------------------------------------------------------VSOR0170
CS          FILE STRUCTURE                                              -VSOR0180
CS                                                                      -VSOR0190
CS              RECORD TYPE                          PRESENT IF         -VSOR0200
CS        ------------------------------------      ---------------     -VSOR0210
CS              FILE IDENTIFICATION                  ALWAYS             -VSOR0220
CS              FILE LABEL                           ALWAYS             -VSOR0230
CS              FILE CONTROL                         ALWAYS             -VSOR0240
CS              FILE INTEGER PARAMETERS              ALWAYS             -VSOR0250
CS                                                                      -VSOR0260
CS   **************(REPEAT OVER ALL GROUPS)                            -VSOR0270
CS   *          SOURCE MOMENTS                       ALWAYS             -VSOR0280
CS   ***************                                                    -VSOR0290
C                                                                       -VSOR0300
CS   **************(REPEAT OVER ALL GROUPS)                            -VSOR0310
CS   *          SCALAR UNCOLLIDED FLUX               ISOP.GT.0          -VSOR0320
CS   ***************                                                    -VSOR0330
C                                                                       -VSOR0340
C-----------------------------------------------------------------------VSOR0350
                                                                         VSOR0360
                                                                         VSOR0370
C-----------------------------------------------------------------------VSOR0380
CR          FILE IDENTIFICATION                                         -VSOR0390
C                                                                       -VSOR0400
CL    HNAME,(HUSE(I),I=1,2),IVERS                                       -VSOR0410
C                                                                       -VSOR0420
CW    1+3*MULT=NUMBER OF WORDS                                          -VSOR0430
C                                                                       -VSOR0440
CD    HNAME              HOLLERITH FILE NAME - VARSOR   - (A6)          -VSOR0450
CD    HUSE(I)            HOLLERITH USER IDENTIFICATION  - (A6)          -VSOR0460
CD    IVERS              FILE VERSION NUMBER                            -VSOR0470
CD    MULT               DOUBLE PRECISION PARAMETER                     -VSOR0480
CD                           1- A6 WORD IS SINGLE WORD                  -VSOR0490
CD                           2- A6 WORD IS DOUBLE PRECISION WORD        -VSOR0500
C                                                                       -VSOR0510
C-----------------------------------------------------------------------VSOR0520
                                                                         VSOR0530
                                                                         VSOR0540
```

```
C-------------------------------------------------------------------------VSOR0550
C                                                                         -VSOR0560
CR          FILE LABEL                                                    -VSOR0570
C                                                                         -VSOR0580
CL    DATE,USER,CHARGE,CASE,TIME,(TITL(I),I=1,12)                         -VSOR0590
C                                                                         -VSOR0600
CW    17*MULT=NUMBER OF WORDS                                             -VSOR0610
C                                                                         -VSOR0620
CD    DATE              AS PROVIDED BY TIMER OPTION 4 - (A6)              -VSOR0630
CD    USER              AS PROVIDED BY TIMER OPTION 5 - (A6)              -VSOR0640
CD    CHARGE            AS PROVIDED BY TIMER OPTION 6 - (A6)              -VSOR0650
CD    CASE              AS PROVIDED BY TIMER OPTION 7 - (A6)              -VSOR0660
CD    TIME              AS PROVIDED BY TIMER OPTION 8 - (A6)              -VSOR0670
CD    TITL(I)           TITLE PROVIDED BY USER        - (A6)              -VSOR0680
C                                                                         -VSOR0690
C-------------------------------------------------------------------------VSOR0700
                                                                           VSOR0710
                                                                           VSOR0720
C-------------------------------------------------------------------------VSOR0730
CR          FILE CONTROL                                                  -VSOR0740
C                                                                         -VSOR0750
CD    IGM,NEUT,JM,LM,IMA,MMA,ISM,IMSISM,ISOP,(IDUM(N),N=1,15)             -VSOR0760
C                                                                         -VSOR0770
CW    25=NUMBER OF WORDS                                                  -VSOR0780
C                                                                         -VSOR0790
CD    IGM               NUMBER OF ENERGY GROUPS                         -VSOR0800
CD    NEUT              LAST NEUTRON GROUP                              -VSOR0810
CD                         (IGM IF ALL NEUTRONS, 0 IF ALL GAMMAS)       -VSOR0820
CD    JM                NUMBER OF SECOND-DIMENSION (J) INTERVALS        -VSOR0830
CD    LM                MAXIMUM LENGTH OF MOMENT EXPANSION              -VSOR0840
CD    IMA               MAXIMUM NUMBER OF FIRST-DIMENSION INTERVALS     -VSOR0850
CD    MMA               NUMBER OF BOUNDARY DIRECTIONS                   -VSOR0860
CD    ISM               NUMBER OF I-BOUNDARY SETS                       -VSOR0870
CD    IMSISM            TOTAL NUMBER OF I-INTERVALS, ALL I-SETS         -VSOR0880
CD    ISOP              UNCOLLIDED FLUX FLAG                            -VSOR0890
CD                          0 - NO UNCOLLIDED FLUX RECORDS PRESENT       -VSOR0900
CD                          1 - UNCOLLIDED FLUX RECORDS PRESENT          -VSOR0910
CD    IDUM(I)           ARRAY SET TO 0                                  -VSOR0920
C                                                                         -VSOR0930
C-------------------------------------------------------------------------VSOR0940
                                                                           VSOR0950
                                                                           VSOR0960
C-------------------------------------------------------------------------VSOR0970
CR          FILE INTEGER PARAMETERS                                       -VSOR0980
C                                                                         -VSOR0990
CL    (LMBIG(IG),IG=1,IGM),                                               -VSOR1000
CL   *(IMBIS(IS),IS=1,ISM),(ISET(J),J=1,JM)                               -VSOR1010
C                                                                         -VSOR1020
CW    IGM+ISM+JM=NUMBER OF WORDS                                          -VSOR1030
C                                                                         -VSOR1040
CD    LMBIG(IG)         LENGTH OF MOMENT EXPANSION FOR GROUP IG         -VSOR1050
CD    IMBIS(IS)         NUMBER OF INTERVALS IN ISET IS                  -VSOR1060
CD    ISET(J)           I-SET ASSIGNED TO INTERVAL J                    -VSOR1070
C                                                                         -VSOR1080
C-------------------------------------------------------------------------VSOR1090
```

```
                                                                      VSOR1100
                                                                      VSOR1110
C-----------------------------------------------------------------------VSOR1120
CR          SOURCE MOMENTS                                           -VSOR1130
C                                                                    -VSOR1140
CL    ((SORM(I,L),I=1,IMS),L=1,LMS)                                  -VSOR1150
C                                                                    -VSOR1160
CW    IMS*LMS=NUMBER OF WORDS                                        -VSOR1170
C                                                                    -VSOR1180
C     DO 1 J=1,JM                                                    -VSOR1190
C   1 READ(N)    *LIST AS ABOVE*                                     -VSOR1200
C                                                                    -VSOR1210
CD    SORM              SOURCE BY INTERVAL AND MOMENT INDEX          - VSOR1220
CD    IMS               IMBIS(IS) FOR IS CORRESPONDING TO J          -VSOR1230
CD    LMS               LMBIG(IG)                                    -VSOR1240
C                                                                    -VSOR1250
C-----------------------------------------------------------------------VSOR1260
                                                                      VSOR1270
                                                                      VSOR1280
C-----------------------------------------------------------------------VSOR1290
CR          SCALAR UNCOLLIDED FLUX                                   -VSOR1300
C                                                                    -VSOR1310
CL    (FLUM(1),I=1,IMS)                                              -VSOR1320
C                                                                    -VSOR1330
CW    IMS=NUMBER OF WORDS                                            -VSOR1340
C                                                                    -VSOR1350
C     DO 1 J=1,JM                                                    -VSOR1360
C   1 READ(N)    *LIST AS ABOVE*                                     -VSOR1370
C                                                                    -VSOR1380
CD    FLUX               UNCOLLIDED FLUX BY INTERVAL                -VSOR1390
C                                                                    -VSOR1400
C-----------------------------------------------------------------------VSOR1410
                                                                      VSOR1420
                                                                      VSOR1430
                                                                      VSOR1440
                                                                      VSOR1450
      END                                                             VSOR1460
C                                                                    -VRFL0010
                                                                      VRFL0020
C**********************************************************************VRFL0030
C                       REVISED 10 NOV 76                            -VRFL0040
C                                                                    -VRFL0050
CF          VARFLM                                                   -VRFL0060
CE          VARIABLE MESH FLUX MOMENT DATA WITH BOUNDARY FLUXES      -VRFL0070
C                                                                    -VRFL0080
C**********************************************************************VRFL0090
                                                                      VRFL0100
CD                      ORDER OF GROUPS IS BY DECREASING ENERGY       VRFL0110
CD                      I IS THE FIRST-DIMENSION INDEX                VRFL0120
CD                      J IS THE SECOND-DIMENSION INDEX               VRFL0130
CD                      JM=1 FOR 1-DIMENSIONAL GEOMETRY               VRFL0140
                                                                      VRFL0150
```

```
C-------------------------------------------------------------------------VRFL0160
CS          FILE STRUCTURE                                                -VRFL0170
CS                                                                        -VRFL0180
CS              RECORD TYPE                          PRESENT IF           -VRFL0190
CS        ------------------------------------      ---------------       -VRFL0200
CS              FILE IDENTIFICATION                  ALWAYS               -VRFL0210
CS              FILE LABEL                           ALWAYS               -VRFL0220
CS              FILE CONTROL                         ALWAYS               -VRFL0230
CS              FILE INTEGER PARAMETERS              ALWAYS               -VRFL0240
CS              FILE REAL    PARAMETERS              ALWAYS               -VRFL0250
CS                                                                        -VRFL0270
CS   **************(REPEAT OVER ALL GROUPS)                              -VRFL0280
CS   *           FLUX MOMENTS                         ALWAYS               -VRFL0290
CS   *           BOUNDARY DIRECTIONAL FLUX            ALWAYS               -VRFL0300
CS   ***************                                                      -VRFL0310
C                                                                         -VRFL0320
C-------------------------------------------------------------------------VRFL0330
                                                                           VRFL0340
                                                                           VRFL0350
C-------------------------------------------------------------------------VRFL0360
CR          FILE IDENTIFICATION                                           -VRFL0370
C                                                                         -VRFL0380
CL    HNAME,(HUSE(I),I=1,2),IVERS                                         -VRFL0390
C                                                                         -VRFL0400
CW    1+3*MULT=NUMBER OF WORDS                                            -VRFL0410
C                                                                         -VRFL0420
CD    HNAME              HOLLERITH FILE NAME - VARFLM    - (A6)         -VRFL0430
CD    HUSE(I)            HOLLERITH USER IDENTIFICATION   - (A6)         -VRFL0440
CD    IVERS              FILE VERSION NUMBER                            -VRFL0450
CD    MULT               DOUBLE PRECISION PARAMETER                     -VRFL0460
CD                           1- A6 WORD IS SINGLE WORD                   -VRFL0470
CD                           2- A6 WORD IS DOUBLE PRECISION WORD         -VRFL0480
C                                                                         -VRFL0490
C-------------------------------------------------------------------------VRFL0500
                                                                           VRFL0510
                                                                           VRFL0520
C-------------------------------------------------------------------------VRFL0530
C                                                                         -VRFL0540
CR          FILE LABEL                                                    -VRFL0550
C                                                                         -VRFL0560
CL    DATE,USER,CHARGE,CASE,TIME,(TITL(I),I=1,12)                         -VRFL0570
C                                                                         -VRFL0580
CW    17*MULT=NUMBER OF WORDS                                             -VRFL0590
C                                                                         -VRFL0600
CD    DATE               AS PROVIDED BY TIMER OPTION 4 - (A6)           -VRFL0610
CD    USER               AS PROVIDED BY TIMER OPTION 5 - (A6)           -VRFL0620
CD    CHARGE             AS PROVIDED BY TIMER OPTION 6 - (A6)           -VRFL0630
CD    CASE               AS PROVIDED BY TIMER OPTION 7 - (A6)           -VRFL0640
CD    TIME               AS PROVIDED BY TIMER OPTION 8 - (A6)           -VRFL0650
CD    TITL(I)            TITLE PROVIDED BY USER        - (A6)           -VRFL0660
C                                                                         -VRFL0670
C-------------------------------------------------------------------------VRFL0680
                                                                           VRFL0690
                                                                           VRFL0700
```

```
C-------------------------------------------------------------------------VRFL0710
CR          FILE CONTROL                                                 -VRFL0720
C                                                                        -VRFL0730
CD    IGM,NEUT,JM,LM,IMA,MMA,ISM,IMSISM,ISBT,ITER,(IDUM(N),N=1,15)       -VRFL0740
C                                                                        -VRFL0750
CW    25=NUMBER OF WORDS                                                 -VRFL0760
C                                                                        -VRFL0770
CD    IGM               NUMBER OF ENERGY GROUPS                        -VRFL0780
CD    NEUT              LAST NEUTRON GROUP                             -VRFL0790
CD                         (IGM IF ALL NEUTRONS, 0 IF ALL GAMMAS)      -VRFL0800
CD    JM                NUMBER OF SECOND-DIMENSION (J) INTERVALS       -VRFL0810
CD    LM                MAXIMUM LENGTH OF MOMENT EXPANSION             -VRFL0820
CD    IMA               MAXIMUM NUMBER OF FIRST-DIMENSION INTERVALS    -VRFL0830
CD    MMA               NUMBER OF BOUNDARY DIRECTIONS                  -VRFL0840
CD    ISM               NUMBER OF I-BOUNDARY SETS                      -VRFL0850
CD    IMSISM            TOTAL NUMBER OF I-INTERVALS, ALL I-SETS        -VRFL0860
CD    ISBT              I-SET FOR SYSTEM BOUNDARIES                    -VRFL0870
CD    ITER              OUTER ITERATION NUMBER AT WHICH FLUX WAS       -VRFL0880
CD                         WRITTEN                                     -VRFL0890
CD    IDUM(I)           ARRAY SET TO 0                                 -VRFL0900
C                                                                        -VRFL0910
C-------------------------------------------------------------------------VRFL0920
                                                                         VRFL0930
                                                                         VRFL0940
C-------------------------------------------------------------------------VRFL0950
CR          FILE INTEGER PARAMETERS                                      -VRFL0960
C                                                                        -VRFL0970
CL    (LMBIG(IG),IG=1,IGM),                                              -VRFL0980
CL   *(IMBIS(IS),IS=1,ISM),(ISET(J),J=1,JM)                              -VRFL0990
C                                                                        -VRFL1000
CW    IGM+ISM+JM=NUMBER OF WORDS                                         -VRFL1010
C                                                                        -VRFL1020
CD    LMBIG(IG)         LENGTH OF MOMENT EXPANSION FOR GROUP IG        -VRFL1030
CD    IMBIS(IS)         NUMBER OF INTERVALS IN ISET IS                 -VRFL1040
CD    ISET(J)           I-SET ASSIGNED TO INTERVAL J                   -VRFL1050
C                                                                        -VRFL1060
C-------------------------------------------------------------------------VRFL1070
                                                                         VRFL1080
                                                                         VRFL1090
C-------------------------------------------------------------------------VRFL1100
CR          FILE REAL PARAMETERS                                         -VRFL1110
C                                                                        -VRFL1120
CL    (Z(J),J=1,JM1),((R(I,IS),I=1,IM1),IS=1,ISM),                       -VRFL1130
CL   *(ENER(IG),IG=1,IGM),EMIN,ENEUT,EV,DEVDK,EFFK,POWER,                -VRFL1140
CL   *(DUMRL(I),I=1,13)                                                  -VRFL1150
C                                                                        -VRFL1160
CW    JM+IMSISM+ISM+IGM+20=NUMBER OF WORDS                               -VRFL1170
C                                                                        -VRFL1180
CD    Z(J)              J-INTERVAL BOUNDARIES                          -VRFL1190
CD    R(I,IS)           I-INTERVAL BOUNDARIES FOR I-SET I              -VRFL1200
CD    ENER(IG)          TOP ENERGY BOUNDARY OF GROUP IG                -VRFL1210
CD    EMIN              BOTTOM ENERGY BOUNDARY OF GROUP IGM            -VRFL1220
CD    ENEUT             BOTTOM ENERGY BOUNDARY OF GROUP NEUT           -VRFL1230
CD                         (0 IF NEUT=0)                               -VRFL1240
```

```
CD     EV                  EIGENVALUE                                          -VRFL1250
CD     DEVDK               DERIVATIVE OF EV VS. EFFK                           -VRFL1260
CD     EFFK                EFFECTIVE MULTIPLICATION FACTOR                     -VRFL1270
CD     POWER               POWER (WATTS) TO WHICH FLUX IS NORMALIZED           -VRFL1280
CD     DUMRL               ARRAY SET TO 0                                      -VRFL1290
CD     JM1                 JM+1                                                -VRFL1300
CD     IM1                 IMBIS(IS)+1                                         -VRFL1310
C                                                                              -VRFL1320
C-------------------------------------------------------------------------------VRFL1330
                                                                               VRFL1340
                                                                               VRFL1350
                                                                               VRFL1360
                                                                               VRFL1370
C-------------------------------------------------------------------------------VRFL1380
CR           FLUX MOMENTS                                                      -VRFL1390
C                                                                              -VRFL1400
CL     ((FLUM(I,L),I=1,IMS),L=1,LMS)                                           -VRFL1410
C                                                                              -VRFL1420
CW     IMS*LMS=NUMBER OF WORDS                                                 -VRFL1430
C                                                                              -VRFL1440
C      DO 1 J=1,JM                                                             -VRFL1450
C    1 READ(N)    *LIST AS ABOVE*                                              -VRFL1460
C                                                                              -VRFL1470
CD     FLUM                FLUX BY INTERVAL AND MOMENT INDEX                  -VRFL1480
CD     IMS                 IMBIS(IS) FOR IS CORRESPONDING TO J                -VRFL1490
CD     LMS                 LMBIG(IG)                                          -VRFL1500
C                                                                              -VRFL1510
C-------------------------------------------------------------------------------VRFL1520
                                                                               VRFL1530
                                                                               VRFL1540
C-------------------------------------------------------------------------------VRFL1550
CR           BOUNDARY DIRECTIONAL FLUX                                         -VRFL1560
C                                                                              -VRFL1570
CL     ((FIO(M,J),M=1,MMA),J=1,JM), ((FJO(M,I),M=1,MMA),I=1,IMB)               -VRFL1580
C                                                                              -VRFL1590
CW     MMA*(JM+IMA)=NUMBER OF WORDS                                            -VRFL1600
C                                                                              -VRFL1610
CD     FIO                 DIRECTIONAL FLUX OUTGOING BY DIRECTION AND         -VRFL1620
CD                           J-INTERVAL                                        -VRFL1630
CD     FJO                 DIRECTIONAL FLUX OUTGOING BY DIRECTION AND         -VRFL1640
CD                           I-INTERVAL. FJO=0 FOR A 1-D GEOMETRY              -VRFL1650
CD     IMB                 IMBIS(IS) FOR IS CORRESPONDING TO ISBT             -VRFL1660
C                                                                              -VRFL1670
C-------------------------------------------------------------------------------VRFL1680
                                                                               VRFL1690
                                                                               VRFL1700
                                                                               VRFL1710
                                                                               VRFL1720
       END                                                                     VRFL1730
```

```
C                                                                       -BNDS0010
C***********************************************************************BNDS0020
C                        PROPOSED  8 OCT 76                             -BNDS0030
C                                                                       -BNDS0040
CF          BNDRYS                                                      -BNDS0050
CE          MULTIPLE BOUNDARY SOURCE DATA                               -BNDS0060
C                                                                       -BNDS0070
C***********************************************************************BNDS0080
                                                                         BNDS0090
CD                       ORDER OF GROUPS IS BY DECREASING ENERGY         BNDS0100
CD                       I IS THE FIRST-DIMENSION INDEX                  BNDS0110
CD                       J IS THE SECOND-DIMENSION INDEX                 BNDS0120
CD                       JM=1 FOR 1-DIMENSIONAL GEOMETRY                 BNDS0130
CD                       NINTSR=NJNTSR=1 FOR EXTERNAL BOUNDARY DATA,     BNDS0140
CD                          AND Z(1)=R(1)=0 IN THAT CASE                 BNDS0150
                                                                         BNDS0160
C-----------------------------------------------------------------------BNDS0170
CS          FILE STRUCTURE                                              -BNDS0180
CS                                                                      -BNDS0190
CS              RECORD TYPE                          PRESENT IF         -BNDS0200
CS        ------------------------------------     ---------------      -BNDS0210
CS              FILE IDENTIFICATION                  ALWAYS             -BNDS0220
CS              FILE LABEL                           ALWAYS             -BNDS0230
CS              FILE CONTROL                         ALWAYS             -BNDS0240
CS              MESH DESCRIPTION                     ALWAYS             -BNDS0250
CS                                                                      -BNDS0260
CS   **************(REPEAT OVER ALL GROUPS)                            -BNDS0270
CS   *          I-BOUNDARY DIRECTIONAL SOURCE     NINTSR.GT.0          -BNDS0280
CS   *          J-BOUNDARY DIRECTIONAL SOURCE     NJNTSR.GT.0          -BNDS0290
CS   ***************                                                    -BNDS0300
C                                                                       -BNDS0310
C-----------------------------------------------------------------------BNDS0320
                                                                         BNDS0330
                                                                         BNDS0340
C-----------------------------------------------------------------------BNDS0350
CR          FILE IDENTIFICATION                                         -BNDS0360
C                                                                       -BNDS0370
CL    HNAME,(HUSE(I),I=1,2),IVERS                                       -BNDS0380
C                                                                       -BNDS0390
CW    1+3*MULT=NUMBER OF WORDS                                          -BNDS0400
C                                                                       -BNDS0410
CD    HNAME              HOLLERITH FILE NAME - BNDRYS   - (A6)          -BNDS0420
CD    HUSE(I)            HOLLERITH USER IDENTIFICATION  - (A6)          -BNDS0430
CD    IVERS              FILE VERSION NUMBER                            -BNDS0440
CD    MULT               DOUBLE PRECISION PARAMETER                     -BNDS0450
CD                           1- A6 WORD IS SINGLE WORD                  -BNDS0460
CD                           2- A6 WORD IS DOUBLE PRECISION WORD        -BNDS0470
C                                                                       -BNDS0480
C-----------------------------------------------------------------------BNDS0490
                                                                         BNDS0500
                                                                         BNDS0510
```

```
C-------------------------------------------------------------------------BNDS0520
CR          FILE LABEL                                                   -BNDS0530
C                                                                        -BNDS0540
CL    DATE,USER,CHARGE,CASE,TIME,(TITL(I),I=1,12)                        -BNDS0550
C                                                                        -BNDS0560
CW    17*MULT=NUMBER OF WORDS                                            -BNDS0570
C                                                                        -BNDS0580
CD    DATE              AS PROVIDED BY TIMER OPTION 4 - (A6)             -BNDS0590
CD    USER              AS PROVIDED BY TIMER OPTION 5 - (A6)             -BNDS0600
CD    CHARGE            AS PROVIDED BY TIMER OPTION 6 - (A6)             -BNDS0610
CD    CASE              AS PROVIDED BY TIMER OPTION 7 - (A6)             -BNDS0620
CD    TIME              AS PROVIDED BY TIMER OPTION 8 - (A6)             -BNDS0630
CD    TITL(I)           TITLE PROVIDED BY USER        - (A6)             -BNDS0640
C                                                                        -BNDS0650
C-------------------------------------------------------------------------BNDS0660
                                                                          BNDS0670
                                                                          BNDS0680
C-------------------------------------------------------------------------BNDS0690
CR          FILE CONTROL                                                 -BNDS0700
C                                                                        -BNDS0710
CL    IGM,JM,IMA,MMA,NINTSR,NJNTSR,(IDUM(I),I=1,19)                      -BNDS0720
C                                                                        -BNDS0730
CW    25=NUMBER OF WORDS                                                 -BNDS0740
C                                                                        -BNDS0750
CD    IGM               NUMBER OF ENERGY GROUPS                        -BNDS0760
CD    JM                NUMBER OF SECOND-DIMENSION (J) INTERVALS       -BNDS0770
CD    IMA               NUMBER OF FIRST-DIMENSION INTERVALS            -BNDS0780
CD    MMA               NUMBER OF BOUNDARY DIRECTIONS                  -BNDS0790
CD    NINTSR             NUMBER OF I-BOUNDARY SOURCES                  -BNDS0800
CD    NJNTSR             NUMBER OF J-BOUNDARY SOURCES                  -BNDS0810
CD    IDUM(I)           ARRAY SET TO 0                                 -BNDS0820
C                                                                        -BNDS0830
C-------------------------------------------------------------------------BNDS0840
                                                                          BNDS0850
                                                                          BNDS0860
C-------------------------------------------------------------------------BNDS0870
CR          MESH DESCRIPTION                                             -BNDS0880
C                                                                        -BNDS0890
CL    (Z(J),J=1,NJNTSR),(R(I),I=1,NINTSR)                                -BNDS0900
C                                                                        -BNDS0910
CW    JM+IMA=NUMBER OF WORDS                                             -BNDS0920
C                                                                        -BNDS0930
CD    Z(J)               POSITION FROM WHICH J-BOUNDARY SOURCES TAKEN   -BNDS0940
CD    R(I)               POSITION FROM WHICH I-BOUNDARY SOURCES TAKEN   -BNDS0950
C                                                                        -BNDS0960
C-------------------------------------------------------------------------BNDS0970
                                                                          BNDS0980
                                                                          BNDS0990
```

```
C-------------------------------------------------------------------------BNDS1000
CR          I-BOUNDARY DIRECTIONAL SOURCES                                -BNDS1010
C                                                                         -BNDS1020
CL    (((BIJ(M,J,N),M=1,MMA),J=1,JM),N=1,NINTSR)                          -BNDS1030
C                                                                         -BNDS1040
CW    MMA*JM*NINTSR=NUMBER OF WORDS                                       -BNDS1050
C                                                                         -BNDS1060
CD    BIJ(M,J,N)         SOURCE IN DIRECTION M, INTERVAL J, SET N         -BNDS1070
C                                                                         -BNDS1080
C-------------------------------------------------------------------------BNDS1090
                                                                           BNDS1100
                                                                           BNDS1110
C-------------------------------------------------------------------------BNDS1120
CR          J-BOUNDARY DIRECTIONAL SOURCES                                -BNDS1130
C                                                                         -BNDS1140
CL    (((BJI(M,I,N),M=1,MMA),I=1,IMA),N=1,NJNTSR)                         -BNDS1150
C                                                                         -BNDS1160
CW    MMA*IMA*NJNTSR=NUMBER OF WORDS                                      -BNDS1170
C                                                                         -BNDS1180
CD    BJI(M,I,N)         SOURCE IN DIRECTION M, INTERVAL I, SET N         -BNDS1190
C                                                                         -BNDS1200
C-------------------------------------------------------------------------BNDS1210
                                                                           BNDS1220
                                                                           BNDS1230
                                                                           BNDS1240
                                                                           BNDS1250
      END                                                                  BNDS1260
C                                                                         -ORDS0010
                                                                           ORDS0020
C*************************************************************************ORDS0030
C                         REVISED 10 NOV 76                               -ORDS0040
C                                                                         -ORDS0050
CF          ORDOSW                                                        -ORDS0060
CE          WORKING FILE FOR THE OAK RIDGE DISCRETE ORDINATES SYSTEM      -ORDS0070
C                                                                         -ORDS0080
C*************************************************************************ORDS0090
                                                                           ORDS0100
CD                        SIMPLE CODES CAN FIND GROUP-ORGANIZED CROSS      ORDS0110
CD                           SECTION DATA BY SKIPPING TO RECORD NBRREC.    ORDS0120
CD                        ORDER OF GROUPS IS BY DECREASING ENERGY          ORDS0130
                                                                           ORDS0140
C-------------------------------------------------------------------------ORDS0150
CS          FILE STRUCTURE                                                -ORDS0160
CS                                                                        -ORDS0170
CS              RECORD TYPE                          PRESENT IF           -ORDS0180
CS        ------------------------------------      ---------------       -ORDS0190
CS              FILE IDENTIFICATION                  ALWAYS               -ORDS0200
CS              FILE LABEL                           ALWAYS               -ORDS0210
CS              FILE LENGTH DATA                     ALWAYS               -ORDS0220
CS              FILE INTEGER PARAMETERS              ALWAYS               -ORDS0230
CS              FILE REAL    PARAMETERS              ALWAYS               -ORDS0240
CS              QUADRATURE SET LENGTHS               NQUAD.GT.0           -ORDS0250
```

```
CS   **************(REPEAT OVER NQUAD QUADRATURE SETS)                    -ORDS0260
CS   *           QUADRATURE SET IDENTIFICATION     NQUAD.GT.0             -ORDS0270
CS   *           QUADRATURE DATA                   NQUAD.GT.0             -ORDS0280
CS   ***************                                                       -ORDS0290
C                                                                         -ORDS0300
CS              PRINCIPAL CROSS SECTION DESCRPTN  MTM.GT.0               -ORDS0310
CS              NUCLIDE IDENTIFICATION            MTM.GT.0               -ORDS0320
C                                                                         -ORDS0330
CS   **************(REPEAT OVER IGM ENERGY GROUPS)                       -ORDS0340
CS   *           CROSS SECTION DATA                MTM.GT.0               -ORDS0350
CS   ***************                                                       -ORDS0360
CS                                                                        -ORDS0370
C---------------------------------------------------------------------------ORDS0380
                                                                           ORDS0390
                                                                           ORDS0400
C---------------------------------------------------------------------------ORDS0410
CR          FILE IDENTIFICATION                                           -ORDS0420
C                                                                         -ORDS0430
CL    HNAME,(HUSE(I),I=1,2),IVERS                                         -ORDS0440
C                                                                         -ORDS0450
CW    NUMBER OF WORDS = 1 + 3*MULT                                        -ORDS0460
C                                                                         -ORDS0470
CD    HNAME             HOLLERITH FILE NAME - ORDOSW   - (A6)           -ORDS0480
CD    HUSE(I)           HOLLERITH USER IDENTIFICATION  - (A6)           -ORDS0490
CD    IVERS             FILE VERSION NUMBER                             -ORDS0500
CD    MULT              DOUBLE PRECISION PARAMETER                      -ORDS0510
CD                          1- A6 WORD IS SINGLE WORD                   -ORDS0520
CD                          2- A6 WORD IS DOUBLE PRECISION WORD         -ORDS0530
C                                                                         -ORDS0540
C---------------------------------------------------------------------------ORDS0550
                                                                           ORDS0560
                                                                           ORDS0570
C---------------------------------------------------------------------------ORDS0580
CR          FILE LABEL                                                    -ORDS0590
C                                                                         -ORDS0600
CL    DATE,USER,CHARGE,CASE,TIME,(TITL(I),I=1,12)                         -ORDS0610
C                                                                         -ORDS0620
CW    NUMBER OF WORDS = MULT*17                                           -ORDS0630
C                                                                         -ORDS0640
CD    DATE              AS PROVIDED BY TIMER OPTION 4 - (A6)            -ORDS0650
CD    USER              AS PROVIDED BY TIMER OPTION 5 - (A6)            -ORDS0660
CD    CHARGE            AS PROVIDED BY TIMER OPTION 6 - (A6)            -ORDS0670
CD    CASE              AS PROVIDED BY TIMER OPTION 7 - (A6)            -ORDS0680
CD    TIME              AS PROVIDED BY TIMER OPTION 8 - (A6)            -ORDS0690
CD    TITL(I)           TITLE PROVIDED BY USER        - (A6)            -ORDS0700
C                                                                         -ORDS0710
C---------------------------------------------------------------------------ORDS0720
```

```
                                                                        ORDS0730
                                                                        ORDS0740
C-------------------------------------------------------------------------ORDS0750
CR          FILE LENGTH DATA                                            -ORDS0760
C                                                                       -ORDS0770
CL    NBRINT,NBRREL,NBRREC,IGM,MTM,NQUAD,NUMDM(14)                      -ORDS0780
C                                                                       -ORDS0790
CW    NUMBER OF WORDS = 20                                              -ORDS0800
C                                                                       -ORDS0810
CD    NBRINT            NUMBER OF INTEGER PARAMETERS = 20               -ORDS0820
CD    NBRREL            NUMBER OF REAL    PARAMETERS = 10 + (1+NC)*IGM  -ORDS0830
CD    NBRREC            NUMBER OF FIRST RECORD OF CROSS SECTION DATA    -ORDS0840
CD    IGM               NUMBER OF ENERGY GROUPS                         -ORDS0850
CD    MTM               NUMBER OF NUCLIDES                              -ORDS0860
CD    NQUAD             NUMBER OF QUADRATURE SETS                       -ORDS0870
CD    NUMDM(I)          ARRAY SET TO 0                                  -ORDS0880
C                                                                       -ORDS0890
C-------------------------------------------------------------------------ORDS0900
                                                                        ORDS0910
                                                                        ORDS0920
C-------------------------------------------------------------------------ORDS0930
CR          FILE INTEGER PARAMETERS                                     -ORDS0940
C                                                                       -ORDS0950
CL    IHT,IHS,IHM,NEUT,IADJ,NSCAT,NCHI,NUMIN(13)                        -ORDS0960
C                                                                       -ORDS0970
CW    NUMBER OF WORDS = NBRINT                                          -ORDS0980
C                                                                       -ORDS0990
CD    NEUT              LAST NEUTRON GROUP                              -ORDS1000
CD                         (IGM IF ALL NEUTRONS, 0 IF ALL GAMMAS)       -ORDS1010
CD    IHT               TABLE POSITION OF TOTAL CROSS SECTION           -ORDS1020
CD    IHS               TABLE POSITION OF SELF-SCATTER CROSS SECTION    -ORDS1030
CD    IHM               CROSS SECTION TABLE LENGTH                      -ORDS1040
CD    IADJ              ADJOINT FLAG                                    -ORDS1050
CD                         0- NON-ADJOINT DATA                          -ORDS1060
CD                         1-     ADJOINT DATA                          -ORDS1070
CD    NSCAT             NUMBER OF LEGENDRE COMPONENTS BEYOND 0TH        -ORDS1080
CD    NCHI              NUMBER OF FISSION SPECTRA                       -ORDS1090
CD    NUMIN(1)          ARRAY SET TO 0                                  -ORDS1100
C                                                                       -ORDS1110
C-------------------------------------------------------------------------ORDS1120
                                                                        ORDS1130
                                                                        ORDS1140
C-------------------------------------------------------------------------ORDS1150
CR          FILE REAL PARAMETERS                                        -ORDS1160
C                                                                       -ORDS1170
CL    (ENER(I),I=1 IGM),EMIN,ENEUT,((CHI(I,NC),I=1,IGM),NC=1,NCHI),     -ORDS1180
CL   *  DUMRL(8)                                                       -ORDS1190
C                                                                       -ORDS1200
CW    NUMBER OF WORDS = NBRREL                                          -ORDS1210
C                                                                       -ORDS1220
CD    ENER(IG)          TOP ENERGY BOUNDARY OF GROUP IG                 -ORDS1230
CD    EMIN              BOTTOM ENERGY BOUNDARY OF GROUP IGM             -ORDS1240
```

```
CD     ENEUT              BOTTOM ENERGY BOUNDARY OF GROUP NEUT          -ORDS1250
CD                           (0 IF NEUT=0)                            -ORDS1260
CD     CHI(I,NC)          FRACTIONAL FISSION YIELD BY ENERGY GROUP      -ORDS1270
CD     DUMRL(I)           ARRAY SET TO 0                                -ORDS1280
C                                                                       -ORDS1290
C-----------------------------------------------------------------------ORDS1300
                                                                         ORDS1310
                                                                         ORDS1320
C-----------------------------------------------------------------------ORDS1330
CR           QUADRATURE SET LENGTHS                                     -ORDS1340
C                                                                       -ORDS1350
CL     (MM(MS),MS=1,NQUAD)                                              -ORDS1360
C                                                                       -ORDS1370
CW     NUMBER OF WORDS = MULT*NQUAD                                     -ORDS1380
C                                                                       -ORDS1390
CD     MM(MS)             NUMBER OF DIRECTIONS BY QUADRATURE SET       -ORDS1400
C                                                                       -ORDS1410
C                                                                       -ORDS1420
C-----------------------------------------------------------------------ORDS1430
                                                                         ORDS1440
                                                                         ORDS1450
C-----------------------------------------------------------------------ORDS1460
CR           QUADRATURE SET IDENTIFICATION                              -ORDS1470
C                                                                       -ORDS1480
CL     (QID(I),I=1,2)                                                   -ORDS1490
C                                                                       -ORDS1500
CW     NUMBER OF WORDS = MULT*2                                         -ORDS1510
C                                                                       -ORDS1520
CD     QID(I)             QUADRATURE SET LABEL - (A6)                  -ORDS1530
C                                                                       -ORDS1540
C-----------------------------------------------------------------------ORDS1550
                                                                         ORDS1560
                                                                         ORDS1570
C-----------------------------------------------------------------------ORDS1580
CR           QUADRATURE DATA                                            -ORDS1590
C                                                                       -ORDS1600
CL     (W(M),I=1,MX),(EMU(M),I=1,MX),(ETA(M),I=1,MX)                    -ORDS1610
C                                                                       -ORDS1620
CW     NUMBER OF WORDS = 3*MX                                           -ORDS1630
C                                                                       -ORDS1640
CD     W(M)               WEIGHT BY DIRECTION                          -ORDS1650
CD     EMU(M)             FIRST   DIRECTION COSINE BY DIRECTION        -ORDS1660
CD     ETA(M)             SECOND  DIRECTION COSINE BY DIRECTION        -ORDS1670
CD     MX                 MM(MS) FOR THIS QUADRATURE SET               -ORDS1680
C                                                                       -ORDS1690
C-----------------------------------------------------------------------ORDS1700
                                                                         ORDS1710
                                                                         ORDS1720
```

```
C-------------------------------------------------------------------------ORDS1730
C                                                                        -ORDS1740
CR          PRINCIPAL CROSS SECTION DESCRIPTION                          -ORDS1750
C                                                                        -ORDS1760
CL    (PCSD(IH),IH=1,IHT)                                                -ORDS1770
C                                                                        -ORDS1780
CW    NUMBER OF WORDS = MULT*IHT                                         -ORDS1790
C                                                                        -ORDS1800
CD    PCSD(IH)          PRINCIPAL CROSS SECTION LABELS - (A6)            -ORDS1810
C                                                                        -ORDS1820
C-------------------------------------------------------------------------ORDS1830
                                                                          ORDS1840
                                                                          ORDS1850
C-------------------------------------------------------------------------ORDS1860
CR          NUCLIDE IDENTIFICATION                                       -ORDS1870
C                                                                        -ORDS1880
CL    (NUC(MT),MT=1,MTM)                                                 -ORDS1890
C                                                                        -ORDS1900
CW    NUMBER OF WORDS = MULT*MTM                                         -ORDS1910
C                                                                        -ORDS1920
CD    NUC(MT)           NUCLIDE LABELS - (A6)                            -ORDS1930
C                                                                        -ORDS1940
C-------------------------------------------------------------------------ORDS1950
                                                                          ORDS1960
                                                                          ORDS1970
C-------------------------------------------------------------------------ORDS1980
CR          CROSS SECTION DATA                                           -ORDS1990
C                                                                        -ORDS2000
CL    ((SIG(IH,MT),IH=1,IHP),MT=1,MTM)                                   -ORDS2010
C                                                                        -ORDS2020
CW    NUMBER OF WORDS = IHP*MTM                                          -ORDS2030
C                                                                        -ORDS2040
CD    SIG(IH,MT)        CROSS SECTION DATA BY TABLE POSITION, THEN BY    -ORDS2050
CD                         NUCLIDE.  TABLE POSITIONS CONTAIN - -         -ORDS2060
CD          1   TO IHT-5  ARBITRARY DATA, SPECIFIED BY USER, OR ABSENT   -ORDS2070
CD              IHT-4   FISSION YIELD FRACTION (RECOMMENDED)             -ORDS2080
CD              IHT-3   FISSION CROSS SECTION  (RECOMMENDED)             -ORDS2090
CD              IHT-2   ABSORPTION CROSS SECTION                         -ORDS2100
CD              IHT-1   NEUTRON PRODUCTION CROSS SECTION                 -ORDS2110
CD              IHT-0   TOTAL CROSS SECTION                             -ORDS2120
CD              IHT+1   UPSCATTER FROM GROUP IG+IHS-IHT-1 TO IG         -ORDS2130
CD          . . . . . .   . . . . . .                                     -ORDS2140
CD          . . . . . .   . . . . . .                                     -ORDS2150
CD              IHS-1   UPSCATTER FROM GROUP IG+1 TO IG                 -ORDS2160
CD              IHS     SELF SCATTER FOR GROUP IG                        -ORDS2170
CD              IHS+1   DOWNSCATTER FROM GROUP IG-1 TO IG              -ORDS2180
CD          . . . . . .   . . . . . .                                     -ORDS2190
CD          . . . . . .   . . . . . .                                     -ORDS2200
CD              IHM     DOWNSCATTER FROM GROUP IG+IHS-IHM TO IG          -ORDS2210
CD              IHM+1   UPSCATTER FROM GROUP G TO ALL GROUPS            -ORDS2212
CD                      (PRESENT ONLY IF IHS.GT.IHT+1                    -ORDS2214
```

```
CD                                                                    -ORDS2215
CD                    IHS MAY BE IHT+1 IF UPSCATTER IS ABSENT.        -ORDS2220
CD                    TRANSFERS FROM GROUPS .LT. 1 OR .GT. IGM        -ORDS2230
CD                       ARE 0. POSITIONS .LE. IHT ARE 0 FOR PL       -ORDS2240
CD                       COMPONENTS OTHER THAN 0TH.                   -ORDS2250
CD                    EACH COMPONENT OF A PL SET IS TREATED AS A      -ORDS2260
CD                       SEPARATE NUCLIDE.  THUS, M SETS WOULD        -ORDS2270
CD                       COMPRISE M*NQUAD+M NUCLIDES                  -ORDS2280
C                  IHP    IHM UNLESS IHS.GT.IHT+1, THEN IHM+1         -ORDS2288
C                                                                     -ORDS2290
C---------------------------------------------------------------------ORDS2300
                                                                       ORDS2310
                                                                       ORDS2320
                                                                       ORDS2330
                                                                       ORDS2340
      END                                                              ORDS2350
```

Section 6. Operation at ORNL

6.1. Program Access

The most up-to-date version of DOT IV is kept in on-line data sets at both X-10 and K-25. Catalogued procedures can be used for access at either location. The user is warned that:

THESE ARE NOT PERMANENT OR QUALITY-ASSURANCE DATA SETS. THEY MAY BE CHANGED WITHOUT NOTICE. JOB FAILURE OR UNSATISFACTORY RESULTS MAY OCCUR.

With this ominous caution in mind, the JCL for access follows:

Example JCL:

```
//DOS PROC PROG=DRIVER,DSNAM='X.WAR14636.PROG',UNT=,VLM=,FT99=SYSIN,     DOS 0010
//  REG=270K,PAR=,SABLK=300,DACYL=5,LGCYL=5,CTC=1064 OR 3458             DOS 0020
//GO EXEC PGM=&PROG,TIME=1440,PARM='EU=-1,&PAR',REGION=&REG              DOS 0030
//STEPLIB DD DSN=&DSNAM,DISP=SHR,UNIT=&UNT,VOL=SER=&VLM                  DOS 0040
//DUMP DD SYSOUT=A,DCB=(RECFM=FA,BLKSIZE=133)                            DOS 0050
//PRINT    DD  SYSOUT=A                                                  DOS 0060
//GO.FT05F001  DD  UNIT=SYSDA,SPACE=(80,(50,25)),                        DOS 0070
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,BUFNO=1)                         DOS 0080
//FT06F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=&CTC)            DOS 0090
//FT07F001 DD SYSOUT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)               DOS 0100
//FT51F001 DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=&CTC)            DOS 0110
//FT53F001 DD DSN=&DSNAM(&PROG),DISP=SHR,VOL=REF=*.STEPLIB               DOS 0120
//GO.FT98F001  DD  UNIT=SYSDA,SPACE=(80,(50,25)),                        DOS 0130
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200,BUFNO=1)                         DOS 0140
//GO.FT99F001  DD  DDNAME=&FT99                                          DOS 0150
//FT01F001  DD  DUMMY                                                    DOS 0160
//FT02F001  DD UNIT=SYSDA,DISP=(NEW,DELETE),                             DOS 0170
// SPACE=(3504,(&SABLK,&SABLK)),DCB=(LRECL=700,BLKSIZE=3504,RECFM=VBS)   DOS 0180
//FT03F001  DD UNIT=SYSDA,DISP=(NEW,DELETE),                             DOS 0190
// SPACE=(3504,(&SABLK,&SABLK)),DCB=(LRECL=700,BLKSIZE=3504,RECFM=VBS)   DOS 0200
//FT04F001  DD UNIT=SYSDA,DISP=(NEW,DELETE),                             DOS 0210
// SPACE=(3504,(&SABLK,&SABLK)),DCB=(LRECL=700,BLKSIZE=3504,RECFM=VBS)   DOS 0220
//FT08F001  DD UNIT=SYSDA,DISP=(NEW,PASS),DSN=&&SIGMAS,                  DOS 0230
// SPACE=(3504,(&SABLK,&SABLK)),DCB=(LRECL=700,BLKSIZE=3504,RECFM=VBS)   DOS 0240
//FT81F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&DACYL,,CONTIG)       DOS 0250
//FT82F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&DACYL,,CONTIG)       DOS 0260
//FT83F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&DACYL,,CONTIG)       DOS 0270
//FT84F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&DACYL,,CONTIG)       DOS 0280
//FT91F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&LGCYL,,CONTIG)       DOS 0290
//FT92F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&DACYL,,CONTIG)       DOS 0300
//FT93F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&DACYL,,CONTIG)       DOS 0310
//FT94F001 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,&LGCYL,,CONTIG)       DOS 0320
```

```
//* EXAMPLES OF PROC USE:                                                    DOS 0330
//* //DOT EXEC DOS,REG=500K  (THIS EXECUTES DOS DRIVER WITH 500K)           DOS 0340
//* //DOT EXEC DOS,PROG=SCALE,FT99=DUMMY  (THIS EXECUTES SCALE DRIVER)      DOS 0350
//   PEND                                                                    DOS 0360
//DOT EXEC DOS                                                                  *//
(  USER SUPPLIED JCL FOR INPUT/OUTPUT DATA SETS  )
=GIP
(  GIP DATA  )
=DOT4P2
(  DOT DATA  )
=END
//
```

6.2. Disk Space Allocation

IBM users are cautioned that each scratch data set record starts on a new track. Thus, for 81:

$$\text{ntrk} = \#\text{ of tracks per record} = 1 + \left[\frac{4*IMA*JM-1}{N}\right]$$

where N is the byte capacity of a track, 7294 for model 2413 disks and 13030 for 3330's, and [] indicates truncation to the next lower integer. Using the following type of JCL for all scratch data sets:

```
//FT81F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),
//    SPACE=(CYL,ncyl,CONTIG)
```

where ncyl is the number of cylinders required to contain ntrk*nrec tracks. The parameter nrec is discussed in a previous section, as is ℓrec, used below.

Using block allocation in the space parameter:

SPACE = (ℓrec, nrec)

may result in job failure. Allocation by track can also cause failure.

The user is also cautioned that ncyl does not always decrease monotonically as NJBLK increases. A safety margin is advised.

6.3. I/Ø Requests and Total Clock Time

The estimation of I/Ø request count and total clock time is probably the most complicated part of DOT IV job submission. If all groups of flux data are contained in core, the number of I/Ø requests is small enough that an estimate based on experience will probably suffice, and the clock time consumed by these I/Ø operations is likewise trivial.

If flux and source data are stored on disk, the clock time can be much larger, depending upon the problem parameters. The following estimates apply:

$$R = R1 + R2 + R3$$

$$T = W + C + T1 + T2 + T3$$

R = total number of I/Ø requests
R1 = requests with all fluxes stored internally
R2 = requests required to compute the inscatter source with fluxes on disk
R3 = requests required to compute the fluxes with fluxes on disk
T = total clock time
T1 = I/Ø time with all fluxes stored internally
T2 = I/Ø time required to compute the inscatter source with fluxes on disk
T3 = I/Ø time required to compute the fluxes with fluxes on disk

The parameters R1 and T1 are typically small and can be estimated from experience. They depend upon the number of energy groups, the number of outer iterations, and other parameters. The value of W depends directly upon the mix of problems sharing the CPU, the data channels, and the disk packs.

The values of R2, R3, T2, and T3 are dominant for large problems. Using the FBSAM[7] disk transmission package, one I/Ø request is logged for each logical record moved, independent of length. The time required depends almost entirely upon the number of tracks written. The logical

units involved are NDFLX, NDSOR, and NDSIN. With fluxes space mesh stored on disk as one block, we can estimate:

$$R2 = 1/2 * IGM*IGM*OI*NJBLK$$
$$R3 = IGM*II*OI*D1*NJBLK$$

The parameter R2 is reduced in the case of cross section sets in which all materials have 0 scattering from certain groups to a given group. In such a case, the I/Ø operations not needed are bypassed.

With the space mesh broken into NJBLK blocks,

$$R2 = IGM*IGM*OI*NJBLK/2$$
$$R3 = IGM*(2+D1+11*II*OI)*NJBLK$$

The time requirements are:

$$T2 = (TM*ntrk+TS)*R2$$
$$T3 = (TM*ntrk+TS)*R3$$

where ntrk is the number of tracks per record, computed previously, and TM and TS are machine-dependent factors. On IBM 2314 disks, TM = 0.040 sec and TS = 0.020 sec. On IBM 3330 disks, TM = 0.034 sec, and TS = -0.017 sec. These factors do not account for cylinder changing, which is not important unless JBLK1 becomes large. The 3330 should be capable of TM = 0.017 sec with TS = 0.009 sec. We are working to obtain this performance.

The user is cautioned that the computer operators at Oak Ridge expect a job to reside in memory no longer than:

$$C + \frac{RM}{1000}$$

where RM = the I/Ø request limit specified on the CLASS card. Thus, the user should always supply, on his class card the limit:

$$RM = ntrk*R$$

REFERENCES

1. ANS Standard "Recommended Programming Practices to Facilitate the Interchange of Digital Computer Programs," prepared by Subcommittee 10, ANS Standards Committee (April 1971).

2. B. M. Carmichael, "Standard Interface Files and Procedures for Reactor Physics Codes, Version III," LA-5486-MS (February 1974).

3. R. Douglas O'Dell, "Standard Interface Files and Procedures for Reactor Physics Codes Version IV," LA-6941-MS (September 1977).

4. W. W. Engle, Jr., "ANISN, A One-Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering," K-1693 (March 1967).

5. W. A. Rhoades and F. R. Mynatt, "The DOT-III Two-Dimensional Discrete Ordinates Transport Code," ORNL-TM-4280 (September 1973).

6. W. W. Engle, M. A. Boling, and B. W. Colston, "DTF-II, A One-Dimensional, Multigroup Neutron Transport Program," NAA-SR-10951 (March 1966).

7. W. A. Rhoades, "The FBSAM Data Transmission Package for IBM 360/370 Computers," ORNL-TM-5199 (January 1976).

8. J. P. Jenal, P. J. Erickson, W. A. Rhoades, D. B. Simpson, and M. L. Williams, "DOQDP, The Generation of a Computer Library for Discrete Ordinates Quadrature Sets, Part I: Fully Symmetric Quadratures and the DOQDP Computer Code, Part II: Common Quadrature Library," ORNL/TM-6023 (September 1977).

INTERNAL DISTRIBUTION

1-3. L. S. Abbott
4. D. E. Bartine
5. D. G. Cacuci
6. R. L. Childs
7. J. D. Drischler
8. M. B. Emmett
9. W. W. Engle, Jr.
10. G. F. Flanagan
11. N. M. Greene
12. H. E. Knee
13. R. A. Lillie
14. D. Loyd
15. J. W. McAdoo
16. R. E. Maerker
17. F. C. Maienschein
18. J. H. Marable
19. F. R. Mynatt
20. J. V. Pace
21. E. M. Oblow
22-41. W. A. Rhoades
42. D. L. Selby
43. E. T. Tomlinson
44. D. R. Vondy
45. C. R. Weisbin
46. J. T. West
47. R. M. Westfall
48. J. E. White
49. L. R. Williams
50. M. L. Williams
51. R. Q. Wright
52. A. Zucker
53. Herbert Goldstein (consultant)
54. Paul Greebler (consultant)
55. Walter B. Loewenstein (consultant)
56. Robert E. Uhrig (consultant)
57. Richard Wilson (consultant)
58-59. Central Research Library
60. ORNL Y-12 Technical Library
61-62. Laboratory Records Department
63. Laboratory Records ORNL RC
64. ORNL Patent Office
65-82. RSIC

EXTERNAL DISTRIBUTION

83. Dr. David Auton, Defense Nuclear Agency, Attn: RATN, Tactical Nuclear Division, Washington, D.C. 20305
84. D. S. Bost, Atomics International, 8900 DeSoto Ave., Canoga Park, CA 91304
85. Steve Crick, General Electric Company, P. O. Box 5029, Sunnyvale, CA 94086
86. R. K. Disney, Westinghouse Advanced Reactor Division, LMFBR Shielding Design & Analysis, Box 158, Madison, PA 15603
87. James Doyle, General Electric Company, Knolls Atomic Power Laboratory, Box 1072, Schenectady, NY 12301
88. Donald E. Emon, GCFR Branch, Reactor Research & Technology Division, U.S. Department of Energy, Washington, D.C. 20545
89. J. L. Fletcher, General Electric Company, Knolls Atomic Power Laboratory, Box 1072, Schenectady, NY 12301
90. M. Y. Gohar, Applied Physics Division, Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439
91. L. A. Hassler, Babcock and Wilcox, Power Generation Group, P. O. Box 1260, Lynchburg, VA 24505
92. Philip B. Hemmig, Reactor Physics Branch, U. S. Department of Energy, Washington, D.C. 20545

93. R. B. Jones, Sandia Laboratories, P. O. Box 5800, Albuquerque, NM 87115
94. Joseph A. Lenhard, Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge, TN 37830
95. Eric H. Ottewitte, EG & G Idaho, Inc., P. O. Box 1625, Idaho Falls, ID 83401
96. R. Protsik, General Electric Company, P. O. Box 5029, Sunnyvale, CA 94086
97. L. D. Swenson, Hanford Engineering Development Laboratory, P. O. Box 1970, Richland, WA 99352
98. Thomas A. Werner, HTGR Branch, Reactor Research & Technology Division, U.S. Department of Energy, Washington, D.C. 20545
99-100. Technical Information Center, U.S. Department of Energy, Oak Ridge, TN 37830