

WANL-PR-(LL)-010

VOLUME 6

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

ODD-K

A TWO-DIMENSIONAL TRANSPORT CODE

by

R. K. Disney

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

ACKNOWLEDGMENTS

The author wishes to express his appreciation to K. D. Lathrop of the Los Alamos Scientific Laboratory for development of the DDK program upon which the ODD-K program is based, and to G. DeBaryshe, formerly of Westinghouse Astronuclear Laboratory, for his efforts in developing the FLOCOW version of the program. Also, the authors appreciate the guidance provided by Mr. Henry E. Stern, Deputy Manager, Nuclear and Plasma Physics Division, George C. Marshall Space Flight Center, the technical monitor of the contract.

ABSTRACT

This report is Volume 6 of nine volumes of the final report on "Synthesis of Computational Methods for the Design and Analysis of Radiation Shields for Nuclear Rocket Systems." Presented in this volume is a manual for the utilization of the ODD-K program which is the Westinghouse Astronuclear Laboratory (WANL) modification of the Los Alamos Scientific Laboratory (LASL) discrete ordinate transport program, DDK. The ODD-K program numerically solves the Boltzmann transport equation using a multigroup, discrete direction representation of the particle flux in a two-dimension (R-Z, R- θ , or X-Y) geometry description of a reactor.

The ODD-K program differs from the LASL DDK program because the program has been modified to perform photon, as well as neutron, transport analysis, using the intermediate FORTRAN IV data processing code, NAGS, described in Volume 7 of this report.

The ODD-K program provides, in conjunction with the NAGS and DAFT codes, a method for obtaining neutron and photon spatial and energy distributions throughout the reactor system, and spatial, energy, and angular distributions of leakage fluxes at the surface of the reactor system.

The code is presently operational on an IBM 7094 computer under the independent FORTRAN II monitor system. Tape assignments are compatible with the Marshall Space Flight Center (MSFC) IBM 7094 computer system. In addition, the ODD-K program operates under a revised FLOCO assembly program, FLOCOW II, which operates as a FORTRAN II monitor system program with off-line input, off-line output and FORTRAN II or IV tape operations.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
	ABSTRACT	iii
1.0	INTRODUCTION	1
2.0	PROGRAM DESCRIPTION	5
2.1	Geometry	5
2.2	Materials	6
2.3	Nuclear Data	6
2.4	Angular Quadrature	11
2.5	Numerical Solution	14
	2.5.1 Inner Iteration	16
	2.5.2 Outer Iteration	16
3.0	INPUT DATA DESCRIPTION	19
3.1	Card Format and Data	19
	3.1.1 Card Format	20
	3.1.2 Data Input Operations	21
3.2	Input Data	22
3.3	Input Deck Setup	42
3.4	Detailed Data Notes	43
3.5	ODD-K Storage Allocation	47
4.0	OUTPUT DESCRIPTION	53
4.1	Printed Output	53
4.2	Tape Output	58
5.0	PROGRAM OPERATION	61
5.1	Tape Assignments	61
5.2	Sense Switch Operations	61
5.3	Program Stops	63
5.4	Assembly Error Stops	63
6.0	REFERENCES	67
	APPENDIX A - FLOCOW ASSEMBLY PROGRAM	69

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Schematic Diagram of the "Final" Design Method	3
2	ODD-K R-Z Geometry Representation	7
3	ODD-K R-Z Mesh Cell	12
4	ODD-K Discrete Directions	13
5	ODD-K Sample Input Cards	23
6	ODD-K Deck Setup	37
7	Graphical Interpretation of Minimum Mesh Line Requirements in an ODD-K Problem	48
8	ODD-K Program Storage Allocation	50

LIST OF TABLES

<u>Number</u>		<u>Page</u>
1	Recommended ODD-K Quadrature Sets	15
2	Definition of Symbols Used for Computing ODD-K Program Storage	52
3	ODD-K Tape Assignments in MSFC IBSYS Version 13	62

SECTION

1.0 INTRODUCTION

This report is Volume 6 of nine volumes of the final report on "Synthesis of Computational Methods for the Design and Analysis of Radiation Shields for Nuclear Rocket Systems." Presented in this volume is a manual for the utilization of the ODD-K program which is the Westinghouse Astronuclear Laboratory (WANL) version of the Los Alamos Scientific Laboratory (LASL) discrete ordinate transport program, DDK.⁽¹⁾ The ODD-K program numerically solves the Boltzmann transport equation using a multigroup, discrete-direction representation of the particle flux in a two-dimension (R-Z, R- θ , or X-Y) description of a reactor geometry.

The ODD-K program is an integral part of a complete and consistent program system solving two dimensional neutron and photon transport problems. The ODD-K program was developed to provide a discrete ordinate transport method which is compatible (by binary tapes) with FORTRAN II and IV programs and operates under the FORTRAN II monitor system control. The program will solve multi-energy group two-dimensional geometry problems with either zero-return-current, reflective, or periodic boundary conditions. The cross sections can be either P_0 , transport corrected, or P_0 and P_1 with up to full up-and down-scattering matrices. The array dimensions are completely flexible. Because of the efficiency of the FLOCOW II loading technique, approximately 26,000 storage locations are available for data on an IBM 7094 Model II computer with 32,768 memory storage locations. A detailed storage allocation description is presented in Section 3.5. Output options include printed output of eigenvalue, scalar fluxes, summary tables, activity tables, and binary tapes containing either neutron or photon energy multigroup angular and scalar flux data.

The ODD-K program is part of the "final" design method schematically shown in Figure 1. This "final" design method is described in detail in Volume 1 of this report. As indicated in Figure 1, the POINT program (Volume 2) prepares macroscopic neutron and photon cross section data for use in the ODD-K program.

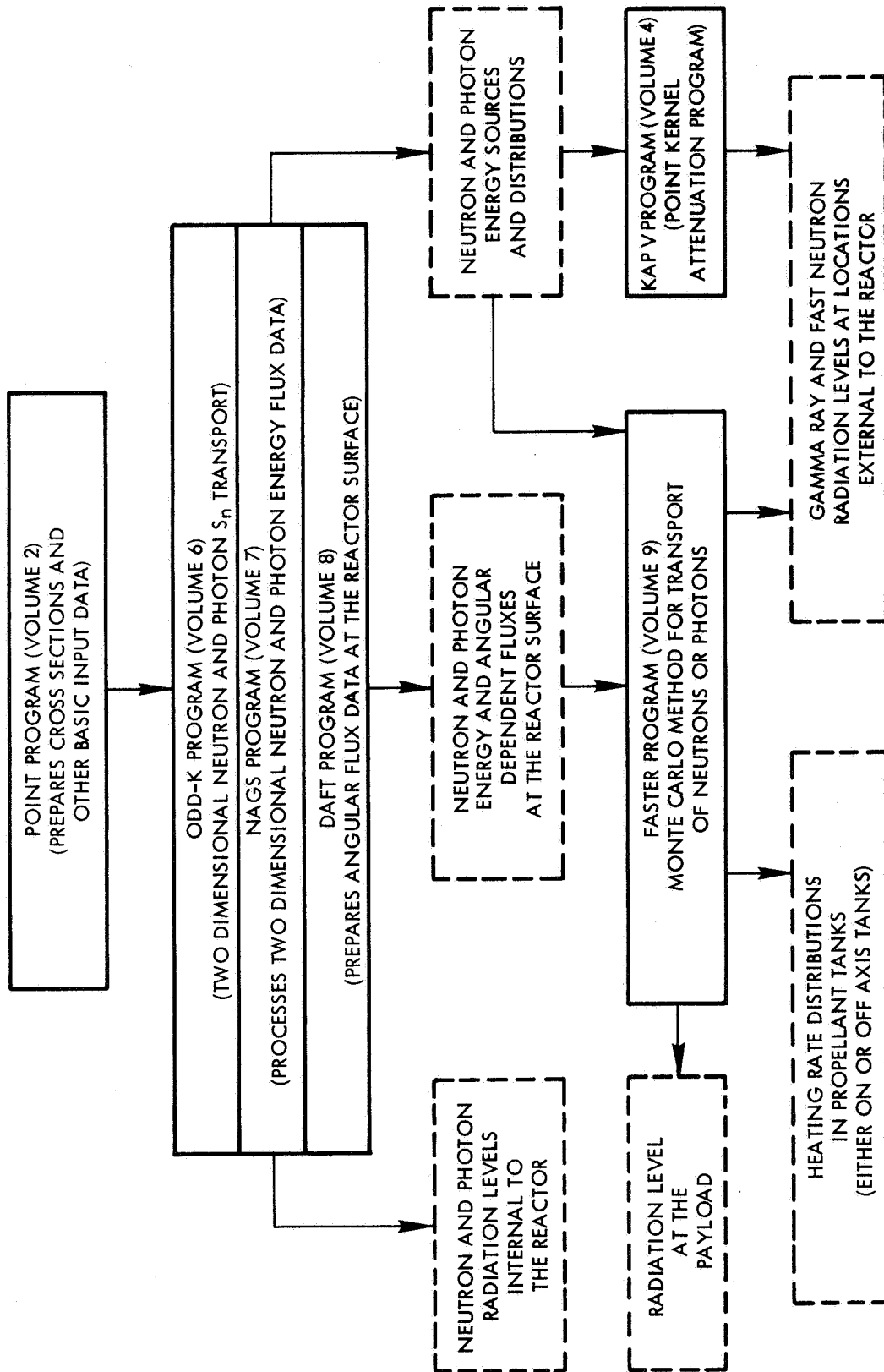
In the "final" design method (See Figure 1), the ODD-K two dimensional transport program provides neutron and photon energy fluxes throughout the reactor geometry. The NAGS data processing program (Volume 7) processes these fluxes and calculates neutron and photon

radiation levels and neutron and photon energy sources within the reactor system. These sources can be employed in either the KAP-V program (Volume 4) or the FASTER Monte Carlo program (Volume 9) for obtaining radiation levels at locations external to the reactor system. In addition, the FASTER program can compute heating rate distributions in the liquid hydrogen propellant (in either an on-axis or an off-axis tank) and the radiation level at the payload. Alternately, the DAFT program (Volume 8) can prepare neutron and photon energy and angular dependent fluxes at the reactor surface from the ODD-K program for use in the FASTER Monte Carlo code.

ODD-K differs from the LASL DDK program because the program has been modified to perform photon, as well as neutron, transport analysis, using the intermediate FORTRAN IV data processing code, NAGS, described in Volume 8 of this report. In addition, ODD-K operates under a revised FLOCO assembly program FLOCOW II, which operates as a FORTRAN II monitor system program with off-line input, off-line output and FORTRAN II or IV tape operations.

The ODD-K code contains the following additional revisions to the LASL DDK code:

- 1) Initial problem normalization with distributed fixed sources was re-programmed to alleviate neutron (or photon energy) balance difficulty.
- 2) A difficulty in the multiplication operations in the mesh cell angular flux solution was eliminated. This decreased problem running times by at least a factor of two.
- 3) Distributed fixed source input was changed to allow this data input as a single logical record on a FORTRAN IV binary tape.
- 4) Scalar and angular flux data output was changed to generate FORTRAN IV binary tapes for use as input to the NAGS and DAFT programs described in Volumes 7 and 8 of this report.
- 5) Parameter and program storage allocation was changed to maximize core memory storage available for input data.
- 6) Modifications were made to allow calculation of various response functions for each mesh cell without lengthy input.



611855-41B

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

The discrete angular segmentation (S_n) method was originally developed by Carlson at LASL as described in detail in Reference 2. The S_n method is basically a numerical, iterative difference method in which the continuous angular distribution of particle flux is solved by considering discrete segmentation of the angular particle direction in a multi-energy group treatment of particle velocity or energy. This method, as programmed in DDK, was implemented as the major portion of the ODD-K program and has been applied to a wide variety of reactor design problems. The modifications of DDK and the use of the FLOCOW II assembly program has improved the utility of the DDK numerical solution in the analysis of neutron and photon transport in complex two-dimensional reactor geometries.

Section 2 presents the geometry and material description used in the ODD-K program. Section 3 describes in detail the input data instructions, deck setup, and tape setups. Output data is discussed in Section 4. The appendix gives details of the FLOCOW assembly program.

SECTION

2.0 PROGRAM DESCRIPTION

The development of the discrete angular segmentation method as originally formulated by Carlson is reported in detail in Reference 2. The various numerical techniques and formal development of the discrete ordinate transport method difference equations as applied in DDK (therefore, in ODD-K), are reported in the referenced document and no attempt will be made to duplicate this work. However, a certain amount of description of the numerical technique and terminology of DDK and ODD-K is required in order to better understand and use the ODD-K program. The following sections provide the physical description used for setup of certain ODD-K problems, and a description of the angular quadrature and the numerical solution procedure used in the ODD-K program.

2.1 GEOMETRY

The description of a reactor geometry in the ODD-K program is performed by a mesh cell overlay of the physical reactor model in a R-Z, R- θ , or X-Y plane. The following discussion, which is limited to the R-Z geometry (see Reference 2 for other geometries), will cover the geometry, material, and nuclear data input requirements of ODD-K.

In the R-Z mesh cell overlay, the reactor is assumed to be uniform in the azimuthal direction. The R-Z plane, as illustrated in Figure 2, is subdivided into IM mesh intervals in the radial (R) direction and into JM mesh intervals in the axial (Z) direction. This mesh interval overlay forms a rectangular mesh cell description of IM x JM mesh cells. Thus, input to the ODD-K program includes the specification of radial and axial mesh coordinates which form the radial and axial boundaries of the IM x JM mesh cells. Each mesh cell is assumed to be of uniform composition. The description of the mesh overlay includes the dimensions and materials in each mesh cell. Solution of scalar fluxes in ODD-K is based on this mesh cell description and the scalar flux is obtained at the mesh cell midpoint, P, as shown in Figure 2.

The determination of mesh coordinate dimensions is dependent on the type of geometry, angular quadrature, and the composition of each mesh cell. Experience in the use of ODD-K has resulted in empirical relationships (Refer to Section 3.5) which are useful in determining mesh cell dimensions. The use of these relationships still requires some intuitive judgement in order to set up an ODD-K problem that will fit within the available computer storage and still obtain a "solution" of the scalar and/or angular fluxes without negative flux values.

2.2 MATERIALS

The materials in an ODD-K problem are specified as uniform material in each mesh cell. The input to the program defines each mesh cell in the $IM \times JM$ overlay as belonging to a specific region. The specific region (i.e., a set of mesh cells), is a representation of a real reactor region. Hence, by specifying the mesh cells comprising a reactor region, the R-Z rectangular representation of this region is described in an ODD-K problem. The material is uniform throughout the region. Each region in the problem is assigned a specific material number. These material numbers correspond to the macroscopic cross section data which are either input to the program by materials or which are calculated internal to the program. In order to maximize the problem size solvable by ODD-K in the normal use of this system of programs, cross section data is calculated in the POINT program (Volume 2 of of this report). The POINT program provides punched decks of macroscopic cross section data for both neutron and photon transport which can be used directly as the material cross section data in the ODD-K regions.

2.3 NUCLEAR DATA

Nuclear data input for ODD-K consists of macroscopic cross section data for materials in the reactor geometry, or microscopic cross section data for the constituents of these materials. The data which are used in the solution of the angular and scalar flux in ODD-K consist of six types of cross sections required for the solution of the multigroup equations and

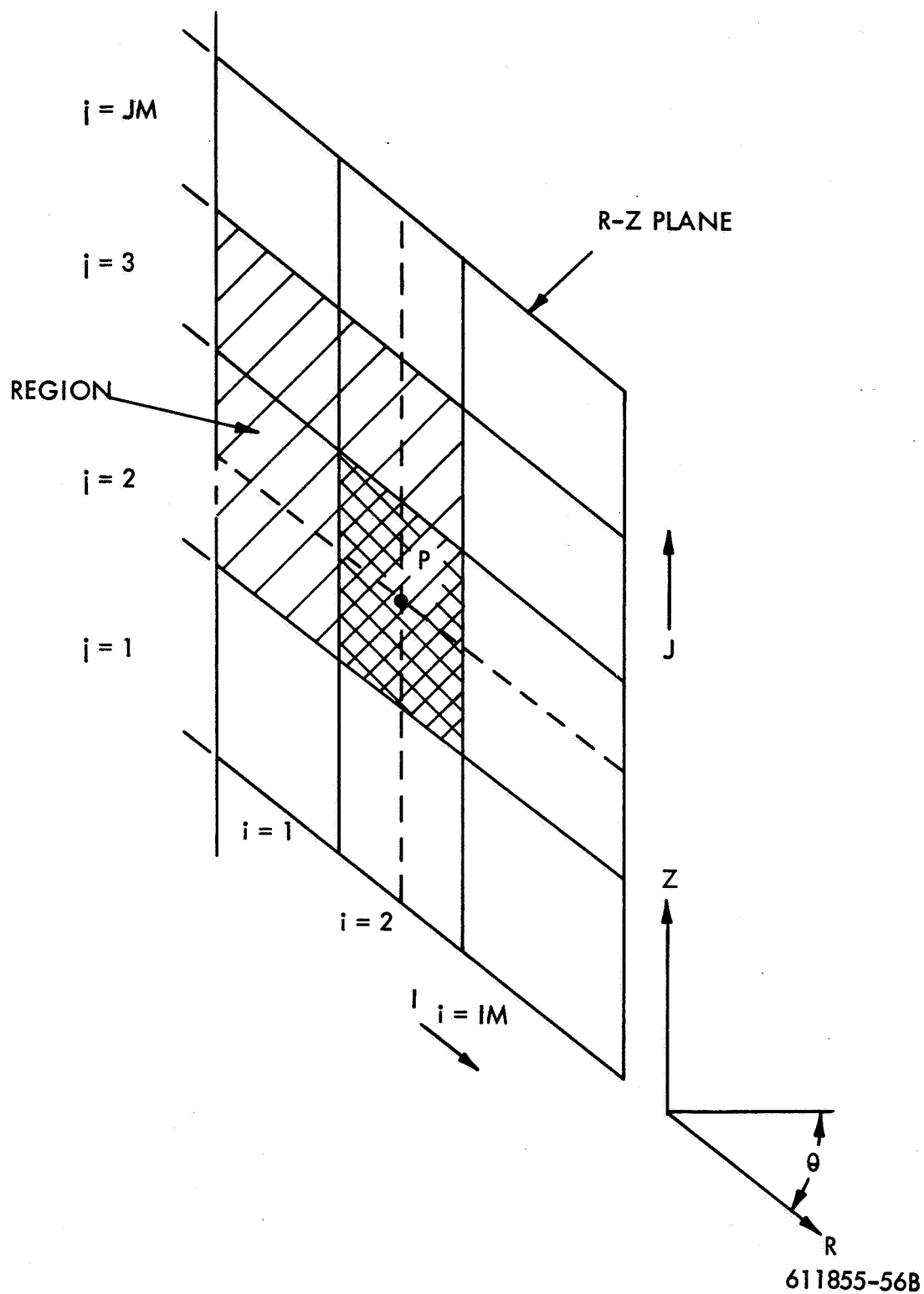


Figure 2. ODD-K R-Z Geometry Representation

further, are used in the balance calculations. These cross section data that are entered for each group, g , in each table of data are:

- 1) The absorption cross section σ_g^a
- 2) The neutron source production cross section, (i.e., the fission cross section, σ_g^f , times the mean number of neutrons per fission, ν_g , in the group, g) $\nu_g \sigma_g^f$
- 3) The total collision (or transport corrected) cross section, σ_g^t
- 4) The upscatter cross sections for scattering into the group g from group g' , where g' is a group of lower energy (higher lethargy), $\sigma_{g' \rightarrow g}^s$ (not used in the MSFC system)
- 5) The within group scatter cross section, $\sigma_{g \rightarrow g}^s$
- 6) The downscatter cross sections for scattering into the group g from group g' where g' is a group of higher energy (lower lethargy), $\sigma_{g' \rightarrow g}^s$

Scattering cross section data for each group is entered as transfer of particles into the group from the groups above or below. A tabular example of the MSFC 16-group neutron cross section library (without upscatter) is as follows:

		Group No., g →						
Cross Section Type, t ↑		1	2	3	4	.	15	16
	1	σ_1^a	σ_2^a	σ_3^a	σ_4^a	.	σ_{15}^a	σ_{16}^a
	2	$\nu\sigma_1^f$	$\nu\sigma_2^f$	.	.	.	.	$\nu\sigma_{16}^f$
	3	σ_1^t	σ_2^t	.	.	.	.	σ_{16}^t
	4	$\sigma_{1 \rightarrow 1}^s$	$\sigma_{2 \rightarrow 2}^s$	.	.	.	.	$\sigma_{16 \rightarrow 16}^s$
	5	0.0	$\sigma_{1 \rightarrow 2}^s$	$\sigma_{2 \rightarrow 3}^s$	$\sigma_{3 \rightarrow 4}^s$	.	.	$\sigma_{15 \rightarrow 16}^s$
	6	0.0	0.0	$\sigma_{1 \rightarrow 3}^s$	$\sigma_{2 \rightarrow 4}^s$	.	$\sigma_{13 \rightarrow 15}^s$	$\sigma_{14 \rightarrow 16}^s$
	7	0.0	0.0	0.0	$\sigma_{1 \rightarrow 4}^s$	.	$\sigma_{12 \rightarrow 15}^s$	$\sigma_{13 \rightarrow 16}^s$
	8	0.0	0.0	0.0	0.0	.	$\sigma_{11 \rightarrow 15}^s$	$\sigma_{12 \rightarrow 16}^s$
	9	0.0	0.0	0.0	0.0	.	$\sigma_{10 \rightarrow 15}^s$	$\sigma_{11 \rightarrow 16}^s$

This library cross section set contains a total of 9 cross section types which are described in detail in Volume 2 of this report. This set does not contain the upscatter cross section data since the use of upscatter in large two dimensional geometry solutions of neutron transport requires excessive computer time and leads to unstable eigenvalue convergence in ODD-K.

In using the ODD-K program for the solution of photon transport problems, upscatter of the photon energy is not required. The standard MSFC photon transport cross sections are 13-group sets with both a P_0 set (isotropic scattering) and a P_1 set (linear anisotropic). The cross sections for the linear anisotropic calculation used in photon transport require two sets of material cross section data. The first set (P_0) is as follows:

	Group No., g →					
	1	2	3	4	.	13
Cross Section Type, t ↑	1	σ_1^a	σ_2^a	σ_3^a	.	σ_{13}^a
	2	0.0	0.0	0.0	.	0.0
	3	σ_1^t	σ_2^t	σ_3^t	.	σ_{13}^t
	4	$\sigma_{1 \rightarrow 1}^{s_o}$	$\sigma_{2 \rightarrow 2}^{s_o}$	$\sigma_{3 \rightarrow 3}^{s_o}$	.	$\sigma_{13 \rightarrow 13}^{s_o}$
	5	0.0	$\sigma_{1 \rightarrow 2}^{s_o}$	$\sigma_{2 \rightarrow 3}^{s_o}$	.	$\sigma_{12 \rightarrow 13}^{s_o}$
	6	0.0	0.0	$\sigma_{1 \rightarrow 3}^{s_o}$	.	$\sigma_{11 \rightarrow 13}^{s_o}$
	7	0.0	0.0	0.0	.	$\sigma_{10 \rightarrow 13}^{s_o}$
	8	0.0	0.0	0.0	0.0	$\sigma_{9 \rightarrow 13}^{s_o}$
	9	0.0	0.0	0.0	0.0	$\sigma_{8 \rightarrow 13}^{s_o}$
	10	0.0	0.0	0.0	0.0	$\sigma_{7 \rightarrow 13}^{s_o}$

The second set, (P_I set), is identical in size except that the cross section types 1 (σ_g^a), 2 ($\nu \sigma_g^f$), and 3 (σ_g^t) are zero and the types 4 through 10 ($\sigma_{g' \rightarrow g}^{s_o}$) are replaced by the P_I cross sections ($\sigma_{g' \rightarrow g}^{s_o}$). The P_I cross section sets must not include the $2I + 1$ multiplication term of the P_I cross sections (i.e., 3.0 times the P_I term). The actual data entered in the photon transport problem are explained in more detail in Volume 2 of this report.

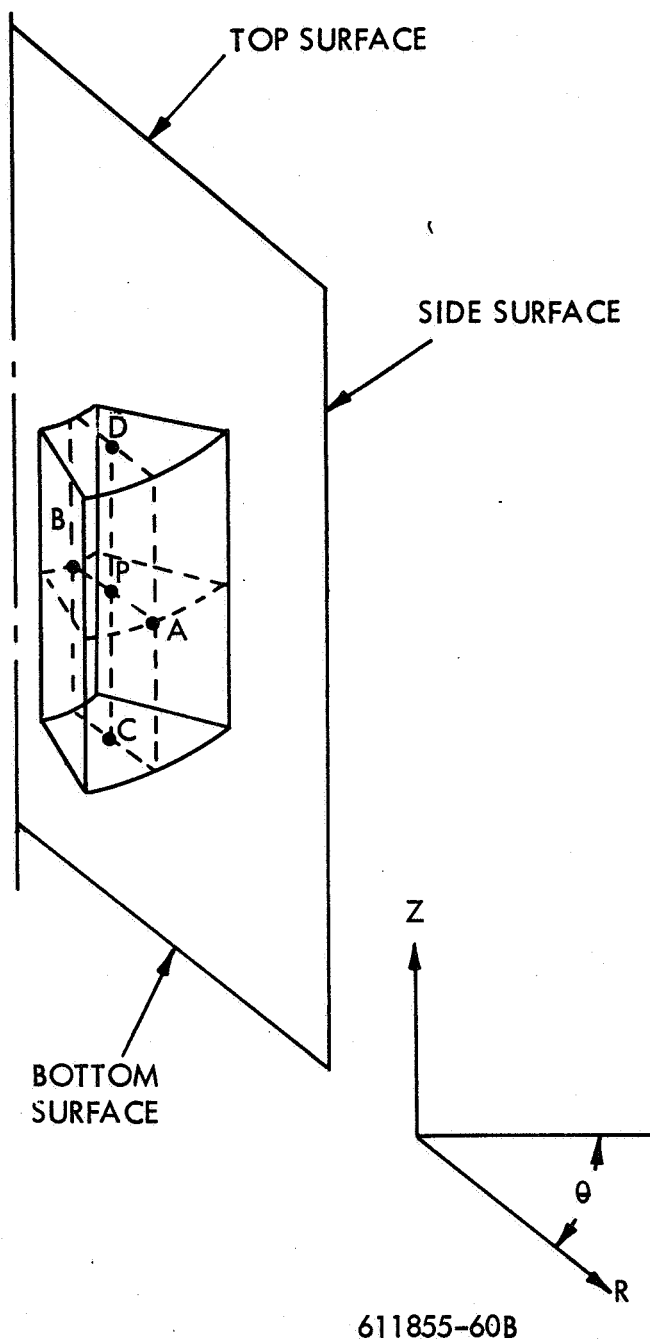
2.4 QUADRATURE DESCRIPTION

The following discussion is limited to: a) a description of the angular fluxes, and b) a recommended discrete ordinate quadrature scheme for an R-Z, ODD-K neutron or photon transport nuclear rocket problem. A description of the numerical solution of angular fluxes in the R-Z, R- θ , and X-Y reactor geometries is presented in Reference 2 and is not covered here. In addition, a detailed description of discrete ordinate quadrature geometry is presented in Reference 3. This discussion should help the reader to understand the angular flux in DDK and in ODD-K.

Angular fluxes are obtained by the ODD-K program at mesh points in each mesh cell in the reactor geometry. These angular fluxes are discrete direction fluxes with directions representative of points on a unit sphere. The discrete direction on the unit sphere is used in the solution of the angular flux. The directions and the weights (finite areas on the unit sphere) are determined by the quadrature geometry scheme employed.

In solving for angular fluxes in an R-Z two-dimensional mesh cell description, each mesh cell is as indicated in Figure 3. The mesh cell is a finite volume element of $\Delta R = R_{i+1} - R_i$, $\Delta Z = Z_{j+1} - Z_j$, and $\Delta \theta = \theta_{k+1} - \theta_k$. Due to symmetry in the angle, θ , point solutions are required only at points A, B, C, D, and P. Points A, B, C, and D are midpoints of each surface of the mesh cell. The angular fluxes obtained at each of these points are then used to calculate the angular and scalar flux at the midpoint of the mesh cell, P. The calculation of scalar flux at P is described in detail in Reference 2. The following discussion provides a description of the ODD-K angular fluxes solved for at the points A, B, C, and D, which are special output data for use in the DAFT code (Volume 8).

At each point (A, B, C, and D in Figure 3) the angular flux is calculated for a hemisphere of the unit sphere centered about each point. The hemisphere about each point is subdivided into its four octants as shown in Figure 4. These octants represent an S_6 order angular quadrature. The numbers in the circles in Figure 4 represent points at which the angular fluxes are obtained in the ODD-K S_6 solution. As shown in the figure, the unit vectors (μ, η, ξ) are represented in an S_6 angular quadrature as direction cosines, $\mu_1, \mu_2, \mu_3, \eta_1, \eta_2, \eta_3, \xi_1,$



611855-60B

Figure 3. ODD-K R-Z Mesh Cell

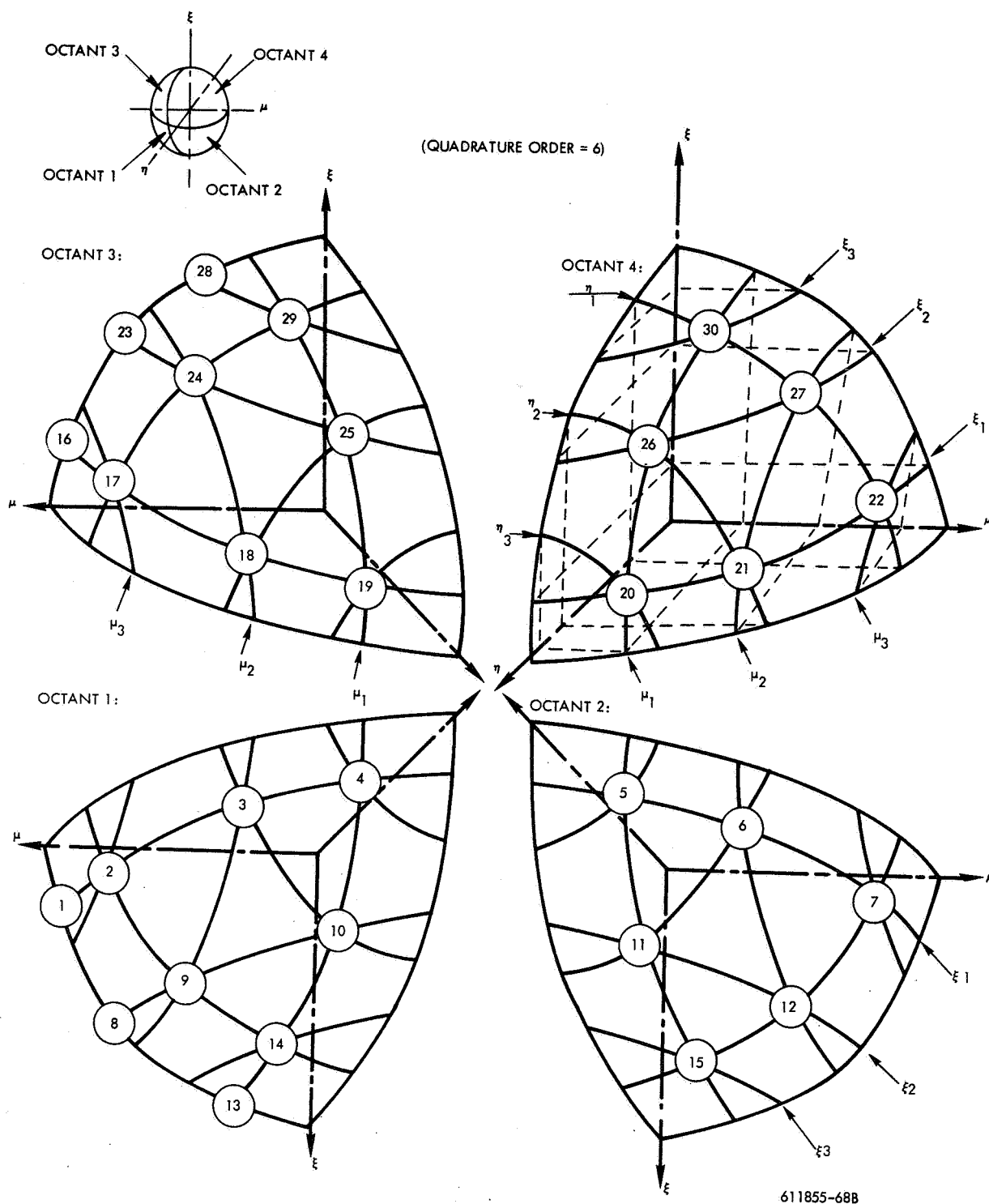


Figure 4. ODD-K Discrete Directions

ξ_2, ξ_3 . With the same distribution of μ_i, η_i, ξ_i on each unit vector, the discrete directions on the surface of the hemisphere lie on latitudes which maintain rotational symmetry in the hemisphere. This rotational symmetry is desirable in the actual solution of angular fluxes.⁽¹⁾ For the purposes of using the ODD-K angular fluxes in the DAFT program, this discrete direction arrangement is required.

The recommended discrete ordinate quadrature input data for the R-Z geometry is the even moment symmetric set as suggested by Lathrop.⁽³⁾ The ODD-K quadrature sets including direction cosines and quadrature weights are presented in Table 1 for quadratures of order 2, 4, and 6. As shown in Figure 4 and presented in Table 1, there are 30 discrete directions in an S_6 ODD-K solution. The numerical solution requires an initialization direction which must be solved in each ξ_i level (e.g., in S_6 : $-\xi_1, -\xi_2, -\xi_3, +\xi_1, +\xi_2, +\xi_3$) in the hemisphere. Each of these initialization directions (indicated as 1, 8, 13, 16, 23, and 28 in Figure 4) are assigned a quadrature weight of zero and do not enter into the scalar flux solution. Therefore, only 24 angular fluxes with non-zero weights are obtained in the S_6 hemisphere. The integration of angular flux in ODD-K to obtain scalar flux is simply the summation of the products of angular flux, times the appropriate quadrature weight.

The angular flux data obtained from the ODD-K program for use in the DAFT code is the mesh cell surface data at the outer radius, top surface and bottom surface of the R-Z reactor geometry. These data, which are obtained as binary tape output (see tape description in Section 4.2), are the angular flux data at point A for all the outer reactor radius mesh cells, C, for all the top surface mesh cells, and D, for all the bottom surface mesh cells.

2.5 NUMERICAL SOLUTION

The numerical solution of the multigroup discrete ordinate difference equations is carried as a two-level iterative scheme. The two types of iterations performed are the inner and the outer iteration. The inner iterations are the group iterations used to solve for the angular flux in all mesh cells for a particular group. An outer iteration is the iteration used to solve the complete set of group equations. From the comparison of sets of successive outer iterations, the eigenvalue and convergence can be evaluated.

TABLE 1
RECOMMENDED ODD-K QUADRATURE SETS

Direction Cosines			Weights
μ_m (M7)	(S_2)	ξ_m (M5)	W_m (W0)
1. -1.000		-0.57735	0.0
2. -0.57735		-0.57735	0.250
3. +0.57735		-0.57735	0.250
4. -1.0000		+0.57735	0.0
5. -0.57735		+0.57735	0.250
6. +0.57735		+0.57735	0.250
μ_m (M7)	(S_4)	ξ_m (M5)	W_m (W0)
1. -0.9367418		-0.3500212	0.0
2. -0.8688903		-0.3500212	0.0833333
3. -0.3500212		-0.3500212	0.0833333
4. +0.3500212		-0.3500212	0.0833333
5. +0.8688903		-0.3500212	0.0833333
6. -0.4950046		-0.8688903	0.0
7. -0.3500212		-0.8688903	0.0833333
8. +0.3500212		-0.8688903	0.0833333
9. -0.9367418		+0.3500212	0.0
10. -0.8688903		+0.3500212	0.0833333
11. -0.3500212		+0.3500212	0.0833333
12. +0.3500212		+0.3500212	0.0833333
13. +0.8688903		+0.3500212	0.0833333
14. -0.4950046		+0.8688903	0.0
15. -0.3500212		+0.8688903	0.0833333
16. +0.3500212		+0.8688903	0.0833333
μ_m (M7)	(S_6)	ξ_m (M5)	W_m (W0)
1. -0.9637974		-0.2666355	0.0
2. -0.9261808		-0.2666355	0.0440315
3. -0.6815076		-0.2666355	0.0393017
4. -0.2666355		-0.2666355	0.0440315
5. +0.2666355		-0.2666355	0.0440315
6. +0.6815076		-0.2666355	0.0393017
7. +0.9261808		-0.2666355	0.0440315
8. -0.7318110		-0.6815076	0.0
9. -0.6815076		-0.6815076	0.0393017
10. -0.2666355		-0.6815076	0.0393017
11. +0.2666355		-0.6815076	0.0393017
12. +0.6815076		-0.6815076	0.0393017
13. -0.3770795		-0.9261808	0.0
14. -0.2666355		-0.9261808	0.0440315
15. +0.2666355		-0.9261808	0.0440315
16. -0.9637974		+0.2666355	0.0
17. -0.9261808		+0.2666355	0.0440315
18. -0.6815076		+0.2666355	0.0393017
19. -0.2666355		+0.2666355	0.0440315
20. +0.2666355		+0.2666355	0.0440315
21. +0.6815076		+0.2666355	0.0393017
22. +0.9261808		+0.2666355	0.0440315
23. -0.7318110		+0.6815076	0.0
24. -0.6815076		+0.6815076	0.0393017
25. -0.2666355		+0.6815076	0.0393017
26. +0.2666355		+0.6815076	0.0393017
27. +0.6815076		+0.6815076	0.0393017
28. -0.3770795		+0.9261808	0.0
29. -0.2666355		+0.9261808	0.0440315
30. +0.2666355		+0.9261808	0.0440315

2.5.1 Inner Iteration

The actual calculation in DDK (ODD-K) proceeds from an input guess of group fluxes. The fission source, distributed fixed source, and the in-scattering into the group are assumed "fixed" within a group solution. The procedure used in DDK to solve the group fluxes is based on a set pattern in the solution of angular and scalar flux. The group calculation (an inner iteration) begins at the top-outermost boundary mesh cell (IM, JM). The top row ($j = JM$) angular fluxes are obtained by applying the top and right boundary conditions and solving the inward and downward discrete directions ($\mu_m < 0.0$ and $\xi_n < 0.0$) in an inward pass through the j th row mesh cells. The left boundary condition is applied using the inward angular flux data for the innermost mesh cell ($i = 1$). Then all outward and downward discrete directions are calculated in an outward pass through the j th row mesh cells. The calculation then proceeds to the next row ($j = JM - 1$), and angular fluxes are calculated in a similar manner using the angular flux data for the row above ($j = JM$). This procedure is followed to the $j = 1$ row at which time the bottom boundary condition is applied to calculate the upward angular flux at $j = 1$. The upward angular flux at row $j = 1$ is then used to calculate the upward discrete direction fluxes in a manner similar to the downward angular flux solution. The calculation proceeds from row $j = 1$ to row $j = JM$ always using the flux data from the $j - 1$ row. The "latest" angular flux data at the left, right, top, and bottom boundaries are always used in the application of boundary conditions. This completes an inner iteration.

2.5.2 Outer Iteration

The group inner iteration convergence is tested by comparing the successive inner iteration self-scattering sources in a group and boundary condition fluxes. Inner iterations are repeated until convergence of group fluxes is achieved or until the input number of inner iterations is reached. The program at the completion of a group solution proceeds to the next group using the "latest" available data for sources into a group. At the completion of all groups, the power (or eigenvalue) is calculated as an integral over all group fluxes. This calculation over all groups completes an outer iteration.

The next calculation is dependent on the type of problem being solved. If the problem is an eigenvalue or parametric eigenvalue calculation, the program proceeds to the next outer iteration using a calculated (scaled) estimate of power (or eigenvalue) as the fission density and scalar flux guess. This procedure is followed for both eigenvalue or parametric eigenvalue calculations, but in parametric eigenvalue calculations the concentrations, dimensions, or fluxes are modified after program-selected outer iterations.

In fixed distributed source calculations without fissions (e.g., photon transport) the angular and scalar flux is obtained with a single outer iteration since particles are not transported from a lower group to a higher group by fission or upscatter. Hence, a single outer iteration yields a solution of photon energy angular and scalar fluxes.

SECTION

3.0 INPUT DESCRIPTION

3.1 CARD FORMAT AND DATA OPERATIONS

The input of data into the ODD-K program is extremely flexible in that data input operations, which are designated by letter symbols, can be used extensively to input arrays or lists of input data in either fixed or floating point form. The standard FLOCOW input format used for all input data (fixed or floating point) and program instructions is an eight word card with each field length (i.e., data field) consisting of eight columns. The first column on the card contains a program control character, and the 10th, 19th, 28th, 37th, 46th, 55th, and 64th columns of each data card are left blank.

Data in the ODD-K program is assigned array labels corresponding to parameter data, data arrays, remarks data, or program instructions. These array labels are assigned in the coding of the program. The user must specify the correct array label to enter data into the proper array.

Parameter data is identified by SXX labels, where S is a letter of the alphabet and the XX is the actual location in the array. Parameter input A00 through V00 are assigned a given number of locations in the ODD-K program and the user must not enter more data than space is allotted.

Data arrays are identified in a similar fashion in that SX labels are used to designate data arrays, where S is a letter of the alphabet and X is a number between 0 and 7. Each SX array is assigned a dimension based on the input parameters to the program. The order and number of pieces of data to be entered in a given SX array must be accurate since an overload will erase data and cause an ODD-K program malfunction.

Remarks data in the ODD-K program are assigned locations in the program by a 900 through 957 label. All remarks (labels 901 - 957) are entered with the program deck and the only required input is a 900 label remark card which identifies the specific ODD-K problem.

Program instructions are entered as data for a FLOCOW program. These data are actual program instructions which make up routines or control the logical flow of the program.

The sets of instructions which make up routines are assigned labels ranging from 800 to 877 and from 960 to 977. The 800-877 series of programs are mainly DDK routines and the 960-977 series are mainly FLOCOW program routines.

In the normal use of ODD-K, only program instructions which control the logical flow of the program are entered as data. These program instructions are called transition cards. These cards are instructions for the program to transfer to a specific program routine (e.g., 960 or 803) and execute the specified instructions internal to that routine. These cards are necessary to separate data decks, program decks, and initiate the solution of an ODD-K problem.

3.1.1 Card Format

The entry of all types of data into ODD-K is controlled by the control punch in column 1 of each card. It must be noted that if a control punch in column 1 is missing, the FLOCOW program will stop reading data and a standard program stop will terminate the job. The valid control punches used in ODD-K are:

- 1) A 9 in column 1 designates a parameter or data array card
- 2) An R in column 1 designates a remark card
- 3) An I in column 1 designates a program instruction card.

The eight data fields of each card can be used to describe the parameters, data, remarks, or program instruction. The label of an array (parameter, data, etc.) is entered in the last three columns of the eight column field.

There are two control symbols (an asterisk and a program instruction) associated with these labels:

Asterisks in the first column of a data field which contains a parameter, data, remark, or program instruction label causes the FLOCOW system to load the first piece of data or remark following the label in the first location of core memory reserved for this labeled array. All data following are loaded sequentially into core memory until another asterisk is encountered. The asterisk is omitted from the first column of the eight column data fields until the input array is terminated by a data field with an asterisk. Therefore, a data continuation card in ODD-K

must have a 9 in column 1 but data in columns 2 - 9, 11 - 18, etc. Blank data fields, in an ODD-K card are assumed to be the termination of any data card and the program skips to read the next physical card in the deck.

Program instruction control symbols used by ODD-K are unique to the program instruction card (I) and are used in the make up of an ODD-K transition card. The principal control symbol is a 4 in the sixth column of the transition card. The entire data card using this 4 symbol contains a fixed point number (N) in the 3, 4, 5 columns, an asterisk in column 2, and the symbol I in column 1. This combination is used with a routine label (e.g., 960, 803) in the 7, 8, and 9 columns of the card. The FLOCOW operation is transfer to and execute the Nth instruction of the routine as labeled. These cards are transition cards for control of logical flow through the program.

3.1.2 Data Input Operations

Fixed point input data are entered as right adjusted data at the extreme right of each eight column field of a card. Floating point data may be entered in several ways. The recommended way, is to enter the number with a decimal point and a sign in the eight column field. If the number is "outsized" and will not fit in the field, then the number should be scaled and the exponent entered in the next field. This exponent data operation is explained in the following discussion.

Special operations are provided for data cards (9 in column 1) to reduce repetitive entry of data on cards. The operation desired is specified in the last column of an eight column data field and may be one of the following letters: E, I, J, R, S, or Z. The remainder of the operation consists of a fixed point number, N, written immediately to the left of the operation symbol. Following are the data operations attached to each operation symbol.

- E: Exponent
- I, J: Interpolation
- R: Repeat
- S: Skip
- Z: Enter Zeroes.

Four of these operations refer to the last number entered (D_0) preceding the operation symbol which itself may have been the result of a data operation. Two of the data operations (I and J) refer both to D_0 and to the next data entry (D_1) on the input card. The data operation are:

Exponent (E). This operation causes D_0 (the number entered in the preceding eight column field) to be multiplied by 10^N , $N < 38$, and applies to floating point numbers only.

Interpolate (I). This operation can be applied to floating point D_0 's and D_1 's only, and constructs N linear interpolants between D_0 and D_1 . The latter number (D_1) must be defined in one field, i.e., must not be followed by an exponent operation.

Interpolate (J). This case is identical to the one above except that it applies to the case when an exponent is associated with D_1 .

Repeat (R). D_0 is repeated N additional times; hence, there are $N + 1$ entries of D_0 . D_0 may be a fixed or floating point number.

Skip (S). Before D_1 is loaded, N memory core locations are skipped over before entering more data.

Enter Zeroes (Z). N consecutive zeroes are loaded into the fixed or floating point data array.

In all of the above operations except Exponent, N is a positive integer, $1 \leq N < 26000$.

An example of each type of data and data operation is illustrated in Figure 5.

3.2 INPUT DATA

The input data required for ODD-K problems are presented below in table form. The data arrays (i.e., blocks of data) are entered according to the input data card and operation descriptions in Section 3.1. The control punch and symbol for each data array is shown in the first column of the table. Parameter data is entered as a labeled block (e.g., A00) and the data A01 through A07 must follow on the same card or on a continuation card. The FLOCO symbol of each input array is shown and the physical symbol associated with the data word

6

Parameter:

9 * A 0 0 5 4 1 4 - 1

9 * L 0 0 1

9 1

Data:

Repeat: 9 * N 0 0 . 0 0 0 1 9 9 9 R

Skip: 9 * C 0 1 S 0 . 0

9 7 S

J Interpolate: 9 * R 0

9 0 . 0 2 J

Exponent: 9 1 . 0 + 2 E

I Interpolate: 9 * R 0 0 . 0 2 I

9 1 0 0 . 0

Zeros: 9 * K 7 1 3 Z

Remarks:

R * 9 0 0 SAMPLE REMARKS

Instructions:

I * 4 8 0 3

I * 5 4 9 6 0

Note: Card columns 28 - 80 are not shown but are considered as blanks in these examples

Figure 5. ODD-K Sample Input Cards

(if any) is presented in the description of the input piece or pieces of data. The arrangement of all input data in a complete ODD-K problem deck is illustrated in Section 3.3. The discussion in this section is in the same order as the actual deck. The tape mountings and dismountings for each type of ODD-K problem are noted herein, but a more detailed description of tapes is included in section 4.0.

The initial input for ODD-K is the parameter data and multigroup cross section, flux, and distributed fixed source data. The parameter data are used in determining the data block sizes in core memory and the starting location of each data block. The multigroup data are read from cards or binary tape and processed in the initial ODD-K operation to obtain group dependent cross section, flux, and distributed fixed source binary work tapes. Group dependent means that each set of group data is a single logical type record on a binary work tape.

The initial input required for ODD-K problems is as follows:

<u>Control Punches</u> (Card Columns 1, 2)	<u>Data Block</u> (Card Columns 7, 8, 9)	<u>FLOCO Symbol</u>	<u>Description</u>
Card Type 1	R*	-	Remarks or alphanumeric title of ODD-K problem in card columns 10 through 72.
Card Type 2	9*	A00	Theory and problem geometry specifications (A01 through A07)
		A01	Problem identification number which will be printed on each on-line printer monitor line.
		A02	If A02 = 0: Regular theory solution. If A02 = 1: Adjoint theory solution. If an adjoint calculation is specified, group reversal of multigroup data is performed internally to the program.

<u>Control Punches</u> (Card Columns 1, 2)	<u>Data Block</u> (Card Columns 7, 8, 9)	<u>FLOCO Symbol</u>	<u>Description</u>
		A03	If A03 = 1: Isotropic (P_0) or transport corrected scattering approximation. If A03 = -1: Anisotropic ($P_0 + P_1$) scattering approximation.
		A04	Order of angular quadrature (S_n order) to be used. It must be an even integer. This number determines the number of discrete directions and the number of data in the W0, M7, M5 input data sets. If A04 = 2: S_2 angular quadrature with 6 discrete directions. If A04 = 4: S_4 angular quadrature with 16 discrete directions. If A04 = 6: S_6 angular quadrature with 30 discrete directions. If A04 = N: S_n angular quadrature with $\frac{N*(N+4)}{2}$ discrete directions.
		A05	If A05 = 1: rectangular X, Y geometry If A05 = 2: cylindrical R, Z geometry If A05 = 3: cylindrical R, θ geometry
		A06	Required for radial zone thickness search calculation. IZ is input as the length (No. of pieces of data) in the zone modifier data block, R3, to be used in zone thickness search calculations. Enter zero otherwise.

<u>Control Punches</u> (Card Column 1, 2)	<u>Data Block</u> (Card Columns 7, 8, 9)	<u>FLOCO Symbol</u>	<u>Description</u>
		A07	Required for axial zone thickness search calculation. Input as the length, JZ, (No. of pieces data) in the zone modifier data block, Z3, to be used in zone thickness search calculations. Enter zero otherwise.
Card Type 3	9*	I00	Detailed calculation type specifications.
		I01	Total number, RZ, of zones in the problem. A zone is occupied by a single homogeneous material.
		I02	Total number, IM, of radial intervals.
		I03	Total number, JM, of axial or azimuthal intervals.
		I04	Calculation Type: If I04 = 0: Calculate a problem with fixed distributed source (Q0).* The fixed source, Q0, is input as a binary tape on unit A-4. The NAGS program as described in Volume 7 of this report, prepares a binary tape for input to ODD-K problems of this type. If I04 = 1: Calculate the multiplication constant (k_{eff}) as the eigenvalue. Set I05 and I06 = 0.0. If I04 = 2: Calculate a time absorption (α) problem. The flux is represented by $N = N_0 (-\alpha t)$

* The ODD-K program will accept absolute magnitude fixed sources without normalization, or the program will normalize the integral fixed source over the problem geometry to the input quantity, S01.

Control
Punches
(Card Columns
1,2)

Data
Block
(Card Columns
7, 8, 9)

FLOCO
Symbol

Description

If I04 = 3: Perform a concentration (C) search calculation. The input densities in the mixture specifications will be varied until a concentration is reached which provides the desired eigenvalue. Desired eigenvalue is entered as the input word, S03.

If I04 = 4: Perform a zone thickness (δ) search calculation. Radial and axial zones will be varied in size according to the R3 and Z3 zone modifier data until the dimensions are reached which provide a desired eigenvalue. Desired eigenvalue is entered as the input word, S03.

I05 Standard first guesses of eigenvalue, EV, for each type of calculation to be performed:

<u>Calculation Type:</u>	Enter I05=	Enter I06=
Q0	0.0	0.0
k_{eff}	0.0	0.0
α	0.0 +	0.1
C	0.0 -	0.05
δ	0.0 -	0.1

I06 Eigenvalue modifier, EVM. The recommended input values are entered as shown in I05 description. The second guess of the eigenvalue in a calculation is $EV \pm EV$; the program chooses the correct sign for EVM.

<u>Control Punches</u> (Card Columns 1,2)	<u>Data Block</u> (Card Columns 7, 8, 9)	<u>FLOCO Symbol</u>	<u>Description</u>
		I07	Epsilon (ϵ). This quantity determines the precision of the calculation. Recommended values of ϵ : Neutron Transport $\epsilon = 0.01$ to 0.001 Photon Transport $\epsilon = 0.0001$ to 0.000001
Card Type 4	9*	G00	Group and material cross section specifications
		G01	Number (GM) of particle (neutron or photon) velocities <u>or</u> energy groups.
		G02	Position (HT) of the total (or transport corrected) cross section (σ_g^t) in the cross section tables. Then, the fission cross section ($\nu_g \sigma_g^f$) is assumed to be in the position, HT-1; the absorption cross section (σ_g^a) is assumed to be in position number HT-2, and activity cross sections are assumed to be in position numbers HT-3 to 1. (Refer to Section 2.3 for a tabular example of the position of cross sections in the MSFC cross section libraries.)
		G03	Position (HS) of the within group scattering cross section ($\sigma_{g \rightarrow g}^s$) in the cross section tables. The position or type numbers HT + 1 to HS-1 are assumed to be upscatter cross sections and, if HS = HT + 1 as in MSFC libraries, no upscatter cross sections are in the tables.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)	G04	Total length (number of positions) of cross section table.
		G05	Enter zero
		G06	Enter zero
		G07	Enter zero
Card type 5	9*	M00	Material specifications
		M01	Total length (MX) of mixture vector or mixture specifications, (length of 10, 11, 12 data blocks)
		M02	Number of cross section sets (MR) to be read from cards
		M03	Number of cross section sets (MT) to be read from a master cross section tape
		M04	Total number (MM) of materials (i.e., number of cross section sets read from cards and tape plus the number of mix- tures formed in the mixture specifica- tions.)
		M05	Enter zero
		M06	Enter zero
		M07	Flux input control If M07 = 0: ODD-K program constructs flux input guess from input fission spectrum (K7 table) If M07 = 1: Read in flux from tape unit A3. If M07 = 2: Read in flux from cards (decimal data cards)

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)	S00	Source specifications
Card type 6	9*	S01	Problem normalization factor, NZ. If $S01 \neq 0.0$: The program normalizes the integral source (fixed distributed + fission) to the value NZ. If $S01 = 0.0$: The program uses fixed distributed source without normalization (i. e., absolute magnitude).
		S02	Parameter source specification, PVT. If $S02 = 0$: No parameter source is input to the problem. The program will use PVT as 1.0. If $S02 = 1$: Fixed reactivity (k_o) is input as PVT. If $S02 = 2$: Fixed time absorption (α_o) is input as PVT.
		S03	Source parameter, PV. If $PVT = 0$, $PV = 0.0$ If $PVT = 1$, $PV = 1.0$ or the desired eigenvalue. If $PVT = 2$, $PV =$ the desired α_o.
		S04	Source Indicator, DM If $S04 = 0$: No distributed source is input to the problem. If $S04 = 1$: Fixed distributed source $Q0_{ijg}$ (specified by $EVT = 0$) is input as a binary tape mounted on unit A-4.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)	S05	Enter as zero.
		S06	Enter as zero.
		S07	Enter as zero.
Card type 7	9*	D00	Input-Output control specifications
		D01	Final print indicator, FP. If D01 = 0: No scalar flux print. If D01 = 1: Print final scalar flux. If D01 = 2: Write final scalar flux as binary tape output on unit B-6. If D01 = 3: Print final scalar flux <u>and</u> write final scalar flux as binary tape output on unit B-6.
		D02	Summary balance table indicator (BT) by zone. If D02 = 0: No summary balance table print for each zone. If D02 = 101+1: Summary balance table prints for each zone plus a summary balance table for all zones.
		D03	Special input-output control (01). Enter as zero.
		D04	Activity print (AM) for selected mesh cells. If D04 = 0: no activity print.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)		If D04 = N: The length, N, of table specifying activity calculations for mesh cells described by pairs of I, J mesh cell indices. The indices I, J are specified in the I5 data block.
		D05	Maximum number (OIM) of outer iterations to be run before a partial problem dump is taken and the program proceeds to next iteration. The ODD-K has a programmed limit of outer iterations (15) which, if exceeded, will provide a complete problem output as though convergence within ϵ (I07) was met.
		D06	Number of binary work tape (TN) for flux data: Enter as 665.
		D07	Number of binary work tape (TNN) for flux data: Enter as 1177.
Card type 8	9*	B00	Boundary condition and activity print specifications
		B01	Left boundary (BL) If B01 = 0: Non-reflective (zero return of flux). If B01 = 1: Reflective. If B01 = 2: Periodic.
		B02	Right boundary (BR). Same conditions (0, 1, 2) as B01.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)	B03	Top boundary (BT). Same conditions (0, 1, 2) as B01.
		B04	Bottom boundary (BB). Same conditions (0, 1, 2) as B01.
		B05	If B05 = 0: No activity calculations. If B05 = N: The length, N, of the data table (I6) which describes the position of the activity cross section type, in the cross section table, to be used for activity calculation.
		B06	If B06 = 0: No activity calculations. If B06 = N: The length, N, of the data table (I4) which describes the material numbers to be used for activity calculations.
Card type 9	9*	C00	
		C01	Maximum number (IIM) of inner iterations (Recommended value: 20 to 30).
		C02	Group skip indicator (GS). If C02 = 0: No group skipping is done in multigroup calculations. If C02 = 1: The groups to be skipped are specified in a table of number (B2). This table defines which groups are to be skipped in the multigroup calculations by $B2g = 0$.
		C03	Diamond method limit (DML) used in numerical solution of the difference equations (Recommended value is 0.5).

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)	C04	Lower limit of change in convergence parameter (LLA). (Recommended value is 0.01.)
		C05	Upper limit of change in convergence parameter (HLA). (Recommended value is 0.05.)
		C06	Eigenvalue change limit (EQL). (Recommended value is 0.005.)
		C07	New parameter modifier (NPM) to be used in parametric eigenvalue calculations. C07 = 1.0 for all problems except for α problems when C07 = 0.75. Alpha calculations usually have a steep eigenvalue search curve which leads to oscillations in the alpha values; these oscillations can be damped by using C07 = 0.75, or in rare cases, C07 = 0.50.
Card type 10	9*	L00	Activity calculation and angular flux output specifications.
		L01	If L01 = 0: Do not perform activity calculation by mesh cell. If L02 = 1: Calculate and print activity results by mesh cell.
		L02	Position (HP) of cross section in the cross section table to be used in the activity calculation.
		L03	Normalization constant (PN) to be multiplied by each mesh cell activity value.

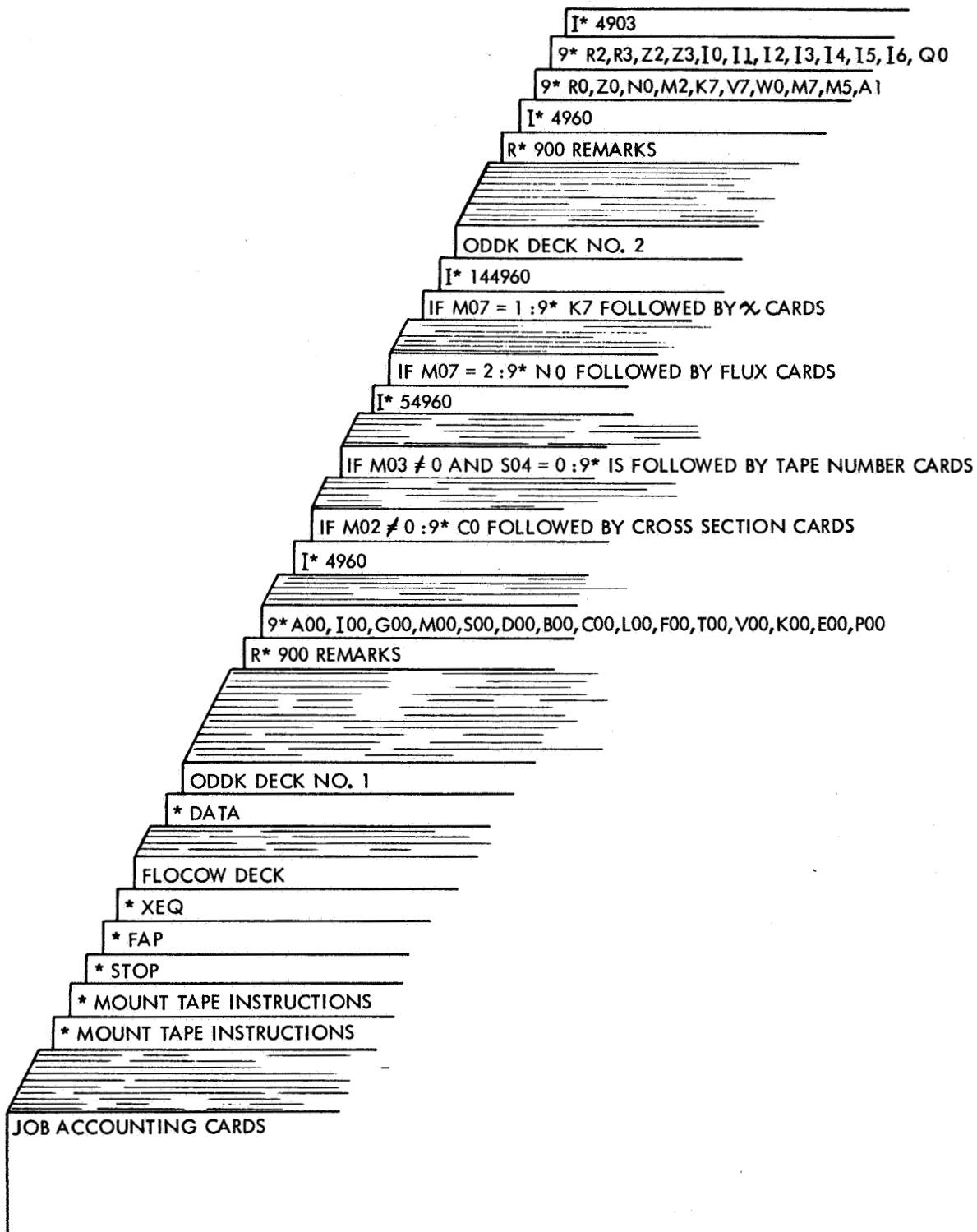
<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)	L04	Angular flux output option. L04 = 0: No angular flux tape output. L04 = 1: Obtain multigroup angular flux tape (described in Section 4.0) on tape unit B-4.
		L05	Enter as zero.
		L06	Enter as zero.
		L07	Enter as zero.

The following cards are parameter space reservation cards which must be input with each problem:

9*	T00	Enter 17 Zeros by 17Z Data Operation
9*	V00	" 47 " by 47Z " "
9*	K00	" 10 " by 10Z " "
9*	P00	Photon transport convergence trigger specification. In order to terminate an ODD-K problem normally after one outer iteration (<u>only one</u> iteration is required for photon or neutron transport with fixed distributed source and no upscatter), a 1 must be entered in P21; hence, a 9* P00 card with 16Z, 1, 10Z, and 1 as data will terminate the problem properly. Enter as 9*, P00, 16Z, 0, 10Z for all other types of calculations.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
9*	E00	E00	Enter 20 zeros by 20Z data operation
1*	4960 (in cols. 6 to 9)		Transition card to transfer to the 960 routine and to execute the routine instructions starting with the 0th instruction.
9*	C0 (If MCR $\neq$ 0)	C0	Cross sections (σ 's) for each material cross section set on cards are entered here. (Decimal data punched cards <u>only</u> .) MCR sets of data.
1*	54960 (in cols. 5 to 9)		Transition card to transfer to the 960 routine and execute the routine instructions starting with the 5th instruction.
9*	N0 (If M07 $\neq$ 0)	N0	Scalar flux data or a flux guess for each mesh cell (IM x JM) for each group. IM x JM x GM values required.
<u>or</u>			
9*	K7 (If M07 $\neq$ 0)	K7	Flux guess is calculated from the x vector, K7. GM (G01) values of required input.
1*	144960 (in cols. 4 to 9)		Transition card to transfer to the 960 routine and to execute the routine instructions beginning with the 14th instruction.

The following data are input following the ODD-K program deck shown in Figure 6. These data, which are the final input to an ODD-K problem, include the geometry, material, and nuclear data required for the numerical solution of the multigroup equations in ODD-K. The data are subdivided into unconditional (always required) and conditional (not always required) input for the types of ODD-K problems. This input is as follows: Unconditional (always required) input for each ODD-K problem.



611855-58B

Figure 6. ODD-K Deck Setup

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)		
R*	900		Remarks or alphanumeric title of the ODD-K problem in columns 10 through 72.
I*	4960 (in cols. 6 to 9)		Transition card to transfer to the 960 routine and to execute the routine instructions beginning with the 0th instruction.
9*	R0		Radial mesh line coordinates. These radial dimensions subdivide the reactor geometry into IM intervals (IM + 1 values).
9*	Z0		Axial (or azimuthal) mesh line coordinates. These axial dimensions subdivide the reactor geometry into JM intervals. (In R- θ problems these values are entered in radians.) (JM + 1 values)
9*	M0		Region or zone identification numbers (fixed point numbers) for each mesh cell in the reactor geometry. These values specify a region as a set of mesh cells. Region identification numbers must be consecutively numbered from 1 to RZ. Values are entered for each radial mesh interval 1-IM for each axial row 1 to JM. IM x JM values.
9*	K7		Neutron fission spectrum. The values, X_g , define the fraction of fission neutrons released in each group. The sum of X_g 's must equal 1.0 for eigenvalue problems. For fixed distributed source problems without fission, all X_g 's are entered as zero. G01 values.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1,2)	(Card Columns 7,8,9)		
9*	V7	V7	Neutron velocities for each group. The value, V_g , defines the average neutron velocity in each group to be used in α calculations. For calcula- tions other than α calculations (including photon transport), all V_g 's are entered as 1.0.

The following input values W0, M5, M7 define the angular quadrature to be used. A description of the quadrature and recommended values are presented in Section 2.2 (Table 1).

9*	W0	W0	Angular quadrature weights. The values W_m are weights of each discrete direction to be used in the integration of angular fluxes. The sum of W_m 's must equal 1.0.
9*	M5	M5	Angular quadrature direction cosines in the unit vector, ξ . The values, ξ_m , define the latitude of each discrete direction on the unit sphere.
9*	M7	M7	Angular quadrature direction cosines in the unit vector, μ . The values, μ_m , define the longitude of each discrete direction on the unit sphere.
9*	A1	A1	Geometry parameters. Three values of A1 are required input: $A1_1 = 1.0$ $A1_2 = 3.14159$ $A1_3 = 3.14159$

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)		
<u>Conditional input</u> required for each type of ODD-K problem:			
9*	R2	R2	If I04 = 4: Then the R2 values must be input. The values of R2 are the region or zone numbers in each mesh cell in a radial row through the fissionable region. These values are used in the selection of zones in a zone thickness search calculation. I02 values.
9*	R3	R3	If I04 = 4: Then the R3 values must be input. The values of R3 are the region or zone modifier parameters to be used in the zone thickness search calculation. I06 values.
9*	Z2	Z2	If I04 = 4: Then the Z2 values must be input. The values of Z2 are the region or zone numbers in each mesh cell in an axial column through the fissionable region. These values are used in the selection of zones for a zone thickness search calculation. I03 values.
9*	M2	M2	Identification numbers (fixed point data) for the material cross section set to be used in each region or zone. Regions or zones are numbered from 1 to RZ. In ODD-K problems with P₁ scattering approximation (e.g., MSFC photon transport), the material numbers are entered as negative values. The program assumes the set specified by M2 to be the P₀ set, and the set M2 + 1 as the P₁ set. Hence, for P₁ calculations there

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)		are <u>twice</u> the number of cross section sets as there are materials.

The following data I0, I1, and I2 are used in forming mixtures of material cross section sets and are also used in the concentration search calculation. (Refer to Section 3.4 for details on use.)

9*	I0		Mixture table numbers (fixed point data) to be used in forming mixtures of material cross section sets. I0's are the mixture identification number. M01 values.
9*	I1		Mixture table numbers (fixed point data) to be used in forming mixtures of material cross section sets. I1's are the material identification numbers for the mixture. M01 values.
9*	I2		Mixture table values (floating point data) to be used in forming mixtures of material cross section sets. I2 is the density (N/cc x 10 ²⁴ or gm/cc) of the materials specified by I1 in the mixture I0. M01 values.
9*	Z3	Z3	If I04 = 4: Then the Z3 values must be input. The values of Z3 are the region or zone modifier parameters to be used in the zone thickness search calculation. A07 values.
9*	I3	I3	If M03 ≠ 0: Then the list of numbers in I3 specify the material or element cross section sets to be read from a master cross section tape which is mounted on unit A3.

<u>Control Punches</u>	<u>Data Block</u>	<u>FLOCO Symbol</u>	<u>Description</u>
(Card Columns 1, 2)	(Card Columns 7, 8, 9)		
9*	I4	I4	If B04 $\neq$ 0: Then the list of numbers I4 specify the material numbers to be used in the activity calculation of specific mesh cells. B04 values.
9*	I5	I5	If D04 $\neq$ 0: Then the list of numbers I5 are the specific mesh cell index pairs (i, j) to be used in the mesh cell activity calculation. D04 values for D04/2 mesh cells.
9*	I6	I6	If B05 $\neq$ 0: Then, the list of numbers I6 specify the cross section type in the cross section table to be used in the specific mesh cell calculation.
9*	Q0	Q0	If I04 = 0 and S04 = 1: Then, the fixed distributed source Q_{ijg} is input to the problem as a binary tape mounted on unit A4. This data must be a single logical tape record of the correct size. IM x JM x GM values.

3.3 INPUT DECK SETUP

The ODD-K problem deck setup is illustrated in Figure 6. This deck includes the FLOCOW program binary object deck, the two ODD-K FLOCO source language program decks, and the input data. The deck order must be as shown in Figure 6. All transition cards must be included. If the conditional data (i.e., N0, K7) is not required as input cards, but is input as binary tape data, then, the transition cards following the cross section data (i.e., C0 or I3) should be the I* 144960 card.

3.4 DETAILED DATA NOTES

This section presents a detailed definition of selected input parameter blocks and data arrays in the ODD-K program. These notes reflect experience in the use of the ODD-K program.

Cross Section Mixing Tables - I0, I1, I2

The cross section mixing table in ODD-K consists of three columns of data (I0, I1, I2) which are used to combine element or material cross section data into macroscopic mixtures of material sets. In addition, these data are used to specify the method of the concentration search calculation. Experience reveals that only imagination limits its flexibility. The following table illustrates the three types of operations performed by the mixing table.

	I0	I1	I2
1)	M	0	X
2)	M	N	X
3)	M	M	0.0

1) Multiply all cross sections in material M by X.

2) Multiply all cross sections in material N by X and add to corresponding cross sections in material M.

3) Multiply all cross sections in material M by the eigenvalue. This is used in concentration search calculation.

Cross Sections - C0

ODD-K requires a table of cross sections for each group, g , of each material in a format described by the input quantities HT, HS, HL, and GM. The position or cross section type in the table is described in general as follows:

<u>Position</u>	<u>Cross Section Type</u>	<u>Meaning</u>
1	σ^{act}	activity
.	.	activity
.	.	activity
.	.	activity
HT - 2	σ^a	absorption
HT - 1	$\nu \sigma^f$	Nu x fission cross section
HT	σ^t	total (or transport corrected)
HT + 1	.	} upscatter
.	.	
.	etc.	
.	$\sigma_{g+2 \rightarrow g}$	
HS - 1	$\sigma_{g+1 \rightarrow g}$	
HS	$\sigma_{g \rightarrow g}$	ingroup scatter
HS + 1	$\sigma_{g-1 \rightarrow g}$	} downscatter
.	$\sigma_{g-2 \rightarrow g}$	
.	$\sigma_{g-3 \rightarrow g}$	
.	etc.	
HL	$\sigma_{g-NDS \rightarrow g}$	

Thus, the parameters HT, HS, and HL completely describe the format of the cross sections. If there are no activity cross sections, HT = 3. If there is no upscatter, HS = HT + 1. If there is no downscatter, HL = HS (i.e., a one group problem).

The P_1 cross section table must correspond in format to the P_0 tables even though the transfer coefficients are the only non-zero numbers. Note: The P_1 cross sections must not contain a $(2I + 1)$ term.

Material Numbers - M0, M2

All cross section sets, whether elements or mixtures, are referred to by a continuous set of material numbers. In particular, the material cross section sets supplied in card form (C0)

become materials 1 through MCR; the materials read from the library tape become MCR + 1 through MCR + MTP, and any number greater than MCR + MTP, but less than or equal to MT refers to a mixture.

When a P_1 cross section set is included with the P_0 set, ODD-K expects the P_1 cross sections to be material number $M + 1$.

Normalization - S01

When the type of calculation in ODD-K is not a distributed fixed source calculation, the total fission source is normalized to S01. When I04 is equal to zero, the total fixed distributed source is normalized to S01 at the beginning of an ODD-K problem, and the fission source, if any, remains unnormalized during iterations. If S01 = 0.0, there is no normalization performed in ODD-K and the input fixed distributed source is unaltered.

Mesh Line Coordinate Dimensions

In order to eliminate (or reduce) negative angular and/or scalar fluxes in ODD-K neutron or photon transport problems, a few simple rules to be used in defining mesh line coordinate dimensions are presented here. When the ODD-K core memory storage limitations prevent the implementing of these guidelines, the techniques to define the minimum necessary mesh coordinate dimensions for ODD-K are illustrated. Negative angular and/or scalar flux solutions may still occur even though these guidelines are followed, but the occurrence of the negative fluxes will be minimized.

The experience obtained in using ODD-K has resulted in the use of the empirical relationships as suggested by Putnam.^(5,6) These criteria, which are based on angular quadrature and region total and scattering cross sections, are considered to be "safe" criteria. The empirical relationships (Criteria 1 and 2) are stringent. A considerable amount of intuitive judgement is required in the use of these criteria.

The radial mesh interval size is approximated by the following equation:

Criteria 1:
$$\Delta R = \frac{1.0 + \frac{\sum_{g \rightarrow g}^{S_0}}{\sum_g^t}}{4.0 \sum_g^t (N - 1)}$$

where: N is the Order of Angular Quadrature
(e.g. S_6 calculation, $N = 6$)

$\sum_g^t$ is the largest total or transport corrected group cross section

$\sum_{g \rightarrow g}^{S_0}$ is the corresponding within group scattering cross section

The axial mesh interval size is approximated by the following equation:

Criteria 2:
$$\Delta Z = \frac{2.0}{\sum_g^t}$$

where: $\sum_g^t$ is the largest total or transport corrected group cross section in the region.

In problems where the storage limitations prevent adherence to the above two guidelines, an intuitive choice of mesh must be made to avoid questionable results. The following procedure should be followed:

Criteria 3: Criteria 1 and 2 should be applied near region boundaries where large flux gradients occur.

Criteria 4: Mesh size should not vary more than a factor of two between adjacent mesh intervals (i.e., Criteria 1 and 2 can be relaxed within a region).

Criteria 5: The intervals near the periphery of the core in R-Z and R- θ problems should follow Criteria 1 and 2.

The remainder of the radial mesh in the core, reflector, etc., can be determined by criteria 4. (See Figure 7 for clarification.)

Justification for usage of criteria 3, 4, and 5 can be based on a before-hand knowledge that most of the particles at any point in a region are produced by sources or scattering down from higher groups rather than direct transport from neighboring points. At the periphery of a reflected reactor, this condition does not exist because of the returning thermal neutrons from the reflector; hence, criteria 1 should be applied at the periphery of the core if negative fluxes are to be avoided.

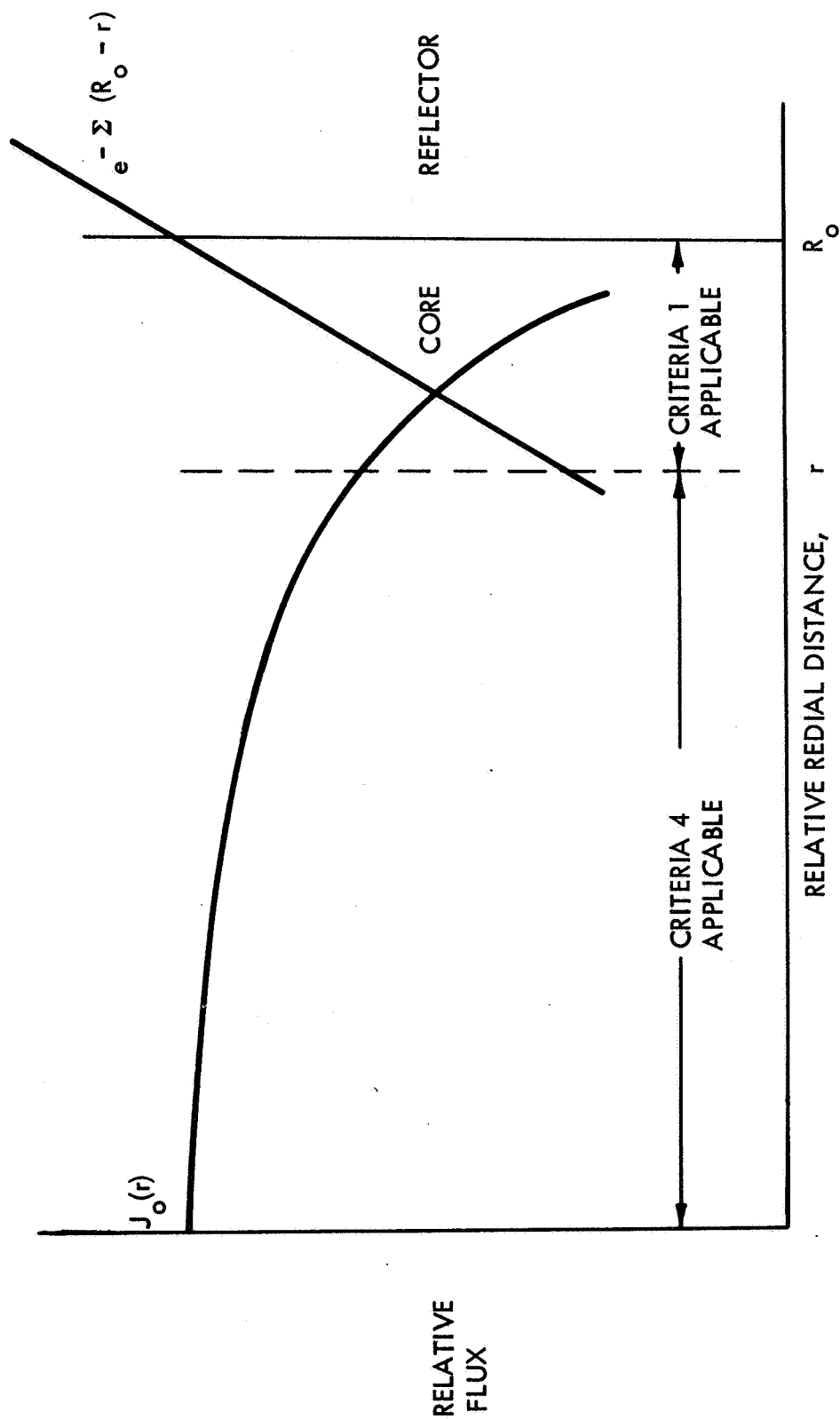
Although negative fluxes may occur using criteria 3, 4, and 5, the location of these negative fluxes and the relative flux level surrounding the negative flux should now have a negligible effect on the overall problem solution. An approximation of mesh interval size in large non-central void regions (such as the void between a reactor and an external shield) may be represented by a method suggested by Putman: "...solutions in non-central voids may be possible by choosing mesh intervals in the void region as if a total cross section of $\Sigma_g^t = 1/\bar{r}$ existed in the region (where $\bar{r}$ is a mean radius). If a large expanse of noncentral void exists, it saves on mesh lines to define several continuous void regions with different r mesh interval sizes to permit fewer mesh intervals in the outer regions where $\bar{r}$ is larger."

Two dimensional neutron transport program calculations using ODD-K have been performed utilizing the recommendations outlined above, and the creditability of the determination of mesh interval sizing has been proven in practice.

3.5 ODD-K STORAGE ALLOCATION

A set of empirical equations which approximately define the size of a problem can be determined from the input data parameters (e.g., number of groups, number of axial and radial intervals, etc.)

The ODD-K program contains two subprograms. Consequently, two sets of conditions must be satisfied when calculating computer memory requirements for an ODD-K problem. These equations are applicable to either (R, θ) or (R, Z) geometries. The first program



NOTE: THE JUNCTION OF CRITERIA 1 AND 4 OCCURS AT THE POSITION WHERE THE REFLECTOR NEUTRONS IN A GROUP CONTRIBUTE A NEGLIGIBLE AMOUNT TO THE TOTAL FLUX IN THAT GROUP.

611855-59B

Figure 7. Graphical Interpretation of Minimum Mesh Line Requirements in an ODD-K Problem

processes input cross section, flux guess, and fixed source data. As a result, the storage limitations are less stringent than for the second program which performs the bulk of the calculations.

Core storage allocation of both subprograms and data information is illustrated in Figure 8. There are 28,321 computer core storage locations available at this point in the ODD-K program for data storage. For a given problem, the storage limitations for the first program will not be exceeded if each of the following three conditions can be met:

$$\sum_{i=1}^3 T_{A_i} < 28321 \quad (\text{condition 1})$$

$$\sum_{i=1}^2 T_{B_i} < 28321 \quad (\text{condition 2})$$

$$T_C < 28321 \quad (\text{condition 3})$$

where:

$$T_{A_1} = 102(103(G01 + 1) + MM), \text{ (If } A03 = 1)$$

$$T_{A_1} = 102(103(G01 + 3) + MM), \text{ (If } A03 = -1)$$

$$T_{A_2} = 103(G01 + 3 + MM), \text{ (If } A03 = -1)$$

$$T_{A_2} = 103(G01 + 1 + MM), \text{ (If } A03 = 1)$$

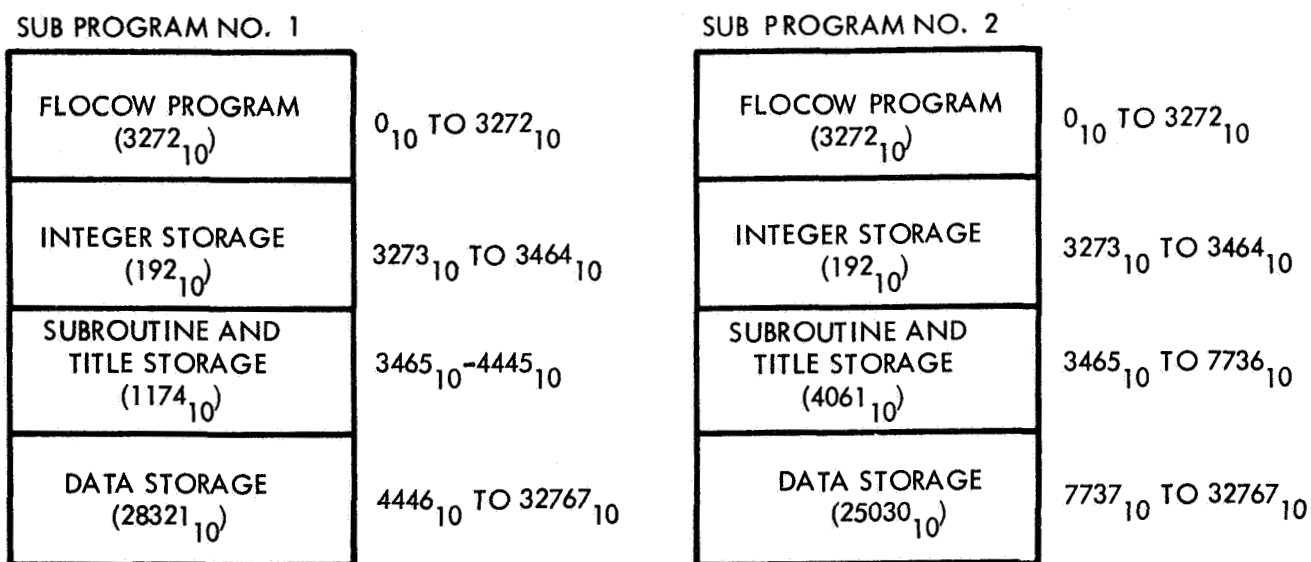
$$T_{A_3} = 2(MM) + 2(G01)$$

$$T_{B_1} = 103(S04 \cdot G01(102 + 1))$$

$$T_{B_2} = (S04 + G05)$$

$$T_C = G04(G01(M04 + 2) + M04 + 1) + M03$$

The three conditions for T_A , T_B , and T_C (above) must each be satisfied for the first program and its data to run on the computer. Symbols used in the above equations are defined in Table 2.



611855-57B

Figure 8. ODD-K Program Storage Allocation (Subscripts refer to a base 10 number system)

Core storage allocation of program and data for the second program of ODD-K is also illustrated in Figure 8. There are 25,030 computer core storage allocations available at this point in an ODD-K problem. For a given problem, the storage limitations for the second program will not be exceeded if the following condition can be met:

$$T_1 + T_2 + T_3 + T_4 + T_5 + T_6 < 25030$$

where:

$$T_1 = 103 [16 + (4 \cdot MM)] , \text{ (If } A03 = 1 \text{ and } B02 + B03 = 0)$$

$$T_1 = 103 [16 + (5 \cdot MM)] , \text{ (If } A03 = 1 \text{ and } B02 + B03 > 0)$$

$$T_1 = 103 [25 + (4 \cdot MM)] , \text{ (If } A03 = -1 \text{ and } B02 + B03 = 0)$$

$$T_1 = 103 [25 + (5 \cdot MM)] , \text{ (If } A03 = -1 \text{ and } B02 + B03 > 0)$$

$$T_2 = 102 [5 + (5 \cdot MM) + 103 (11 + L01)] , \text{ (If } A03 = 1 \text{ and } B02 + B03 = 0)$$

$$T_2 = 102 [5 + (5 \cdot MM) + 103 (12 + L01)] , \text{ (If } A03 = 1 \text{ and } B02 + B03 > 0)$$

$$T_2 = 102 [5 + (5 \cdot MM) + 103 (19 + L01)] , \text{ (If } A03 = -1 \text{ and } B02 + B03 = 0)$$

$$T_2 = 102 [5 + (5 \cdot MM) + 103 (20 + L01)] , \text{ (If } A03 = -1 \text{ and } B02 + B03 > 0)$$

$$T_3 = B05 [2 + (101 + 1) \cdot B06] , \text{ (If } B05 > 0)$$

$$T_4 = [(2 \cdot B06) + M04(G04 + 1) + (4 \cdot D02) + A06 + A07 + (3 \cdot M01)]$$

$$T_5 = [(19 \cdot MM) + (2 \cdot 101) + 26]$$

$$T_6 = G01(16 + (3 \cdot D02)) , \text{ (If } D04 = 0)$$

$$T_6 = [G01(16 + (3 \cdot D02)) + \frac{7 \cdot D04}{2} + (5 \cdot D04)] , \text{ (If } D04 > 0)$$

Symbols used in the above equations are defined in Table 2.

It must be emphasized that, if the conditions described above are satisfied, there is no guarantee that the ODD-K program will fit on the computer. However, experience has shown that the solution to the equations prove to be accurate with ± 75 locations.

Table 2
DEFINITION OF SYMBOLS USED FOR
COMPUTING ODD-K PROGRAM STORAGE

<u>ODD-K Variable</u>		<u>Meaning (Refer to Section 3.1, also)</u>
A03	=	$\left\{ \begin{array}{l} = 1 \text{ Isotropic} \\ = -1 \text{ Antistropic} \end{array} \right.$
A04	=	Order of Angular Quadrature
A06	=	Length of R3 Table
A07	=	Length of Z3 Table
I01	=	Total Number of Zones
I02	=	Number of Radial Intervals
I03	=	Number of Axial Intervals
G01	=	Number of Energy Groups
G04	=	Length per group of the Cross Section Table
M01	=	Length of Mixture Table
M03	=	Number of Cross Section sets from tape
M04	=	Total number of materials including mixtures
D02	=	Number of zones + 1
D04	=	$\left\{ \begin{array}{l} = 0 \text{ No Activity Print} \\ = N \text{ where } N = \text{Length of Activity Table} \end{array} \right.$
B02	=	Right Boundary Condition
B03	=	Left Boundary Condition
B05	=	Length of I6 Table, position of activity type for activity print
B06	=	Length of I4 Table, material numbers for activity print
L01	=	$\left\{ \begin{array}{l} = 0 \text{ Skip Reaction Rate Print} \\ = 1 \text{ Reaction Rate by Mesh Point} \end{array} \right.$
S04	=	$\left\{ \begin{array}{l} = 0 \text{ No fixed source} \\ = 1 \text{ Fixed Source} \end{array} \right.$
MM	=	$(A04 + 4) A04/2$

SECTION

4.0 OUTPUT DESCRIPTION

The output from an ODD-K problem consists of printed output and FORTRAN IV binary tapes. The printed output is obtained as BCD output (logical unit 6, MSFC tape unit B-1) which is printed on the off line printing device (IBM 1401 or 360/30). In addition, a limited amount of on-line printing is obtained from the ODD-K problem during execution. This on-line output defines core memory storage allocation and, under sense switch operation, the monitor line consisting of problem convergence data for each outer iteration. The printed output (on-line and off-line) is described in more detail in Section 4.1.

Binary tape output obtained from ODD-K consists of a scalar flux tape and an angular flux tape. The tape output from ODD-K is described in detail in Section 4.2.

4.1 PRINTED OUTPUT

Printed output is dependent on the type of calculation performed. The following description of output is limited to the portions of the printed data which are not self-explanatory. Optional printed data are noted. The off-line printed output is as follows:

1) If $S04 = 1$: The input values of fixed distributed source is printed as a table of values for each group. The printed format is as follows:

	1	2	3	4	5	6	7	8
1	Q0 for group 1							
2								
3								
.								
IM-1								
IM								
	9	10	11	.	.	JM-1	JM	
1	Q0 for group 1							
2								
3								
.								
IM-1								
IM								

where, IM is the number of radial intervals and JM is the number of axial intervals in each group as shown above.

2) The input and calculated values of material cross sections for all materials (M04 material sets). Printed output is a table of all material cross sections by cross section type for each group. The following example of ODD-K printed output is the MSFC neutron cross section sets as follows:

	1	2	3	4	5	6	7	8
1	σ^a	σ^a						
2	σ^f	σ^f						
3	σ^t	σ^t						
4	$\sigma^s_{1 \rightarrow 1}$	$\sigma^s_{2 \rightarrow 2}$		σ^i 's for group 1				
5	0	0						
6	0	0						
7	0	0						
8	0	0						
9	0	0						

	9	10	12	M04-1	M04
1					
2					
3					
4					
5		σ^i 's for group 1			
6					
7					
8					
9					

Group 2 through G01 follow as above except $\sigma^s_{g^i \rightarrow g}$ appear in cross section types 5 - 9.

3) The input parameter data has eight lines of output. A line for each of the labeled blocks of parameter data, A00, I00, G00, M00, S00, C00, D00, B00, and L00.

4) The input geometry specifications of

R1: Input Quantity R0
 R2: Input values
 R3: Input values
 Z1: Input quantity Z0
 Z2: Input values
 Z3: Input values
 M2: Input values

5) The angular quadrature specifications of

M4: Direction cosine mates of M5 input values
 M5: Input values
 M6: Direction cosine mates of M7 input values
 M7: Input values
 W0: Input values
 W1: $W0 * M7$
 W2: $W0 * M5$

6) The input data blocks of I0, I1, I2, I6, I4, I5, K7, and V7.

7) Apictorial representation of the mesh cell description by region where region

numbers are represented by the following symbolism:

<u>Region No.</u>	<u>Symbol</u>	<u>Region No.</u>	<u>Symbol</u>
1	1	19	J
2	2	20	K
3	3	21	L
4	4	22	M
5	5	23	N
6	6	24	O
7	7	25	P
8	8	26	Q
9	9	27	R
10	A	28	S
11	B	29	T
12	C	30	U
13	D	31	V
14	E	32	W
15	F	33	X
16	G	34	Y
17	H	35	Z
18	I		

Problems with more than 35 regions will not print properly since the 36th, 72nd, etc. region symbols cannot be printed, and a distorted printout will be obtained.

8) A pictorial representation of the mesh cell description by material, where material numbers are represented by the above symbolism.

9) A line of convergence data for each outer iteration. This line contains the following:

A01 Problem identification number

ICT Outer iteration count

LCT Cumulative inner iteration count

NB System particle balance for each outer iteration

$$NB = \frac{\text{System leakage} + \text{absorption} + \text{out scatter}}{\text{Fixed distributed source} + \text{fissions} + \text{in scatter}}$$

EPS Input convergence criterion

EQ Eigenvalue change in search calculation

EV Eigenvalue

LA Convergence parameter, λ

10) A system particle balance table including a line for each group and a line for the total over-all groups. The separate quantities printed out are:

Fixed Source: $Q_0^T = \sum_{ij} Q_{0,ij} V_{0,ij}$

Fission Source: $F_0^T = \left(\frac{X_g}{k_{eff}} \right) \times \left(\sum_{ij} F_{0,ij} V_{0,ij} \right)$

Zone Number: the region or zone number in the geometry

Total Particle Flux: $N_g = \sum_{ij} N_{2,ij} V_{0,ij}$

Total Particle Density: $ND_g = N_g / V_g$

Fission Neutrons: $FN_g = \sum_{ij} N_{2,ij} V_{0,ij} \sum_g^f$

11) A final monitor line of convergence data as described in item 9.

12) Final geometry specifications and radial and axial fission densities

R1: Radial mesh line coordinate dimensions

R2: Radial mesh interval midpoint dimensions

Z1: Axial mesh line coordinate dimensions

Z2: Axial mesh interval midpoint dimensions

M2: Input values

$$F0_i^T: \sum_i F0_{ij} A3_i$$

$$F0_i^T: \sum_i F0_{ij} A2_i$$

13) The mesh cell fission density, $F0_{ij}$, as a table of values as follows:

	1	2	3	4	.	.	JM-1	JM
1	$F0_{ij}$ for all mesh cells							
2								
3								
4								
.								
.								
IM-1								
IM								

14) The mesh cell scalar flux, $N0_{ij}$, as a table of values for each group as follows:

	1	2	3	4	.	.	JM-1	JM
1	$N0_{ij}$ for all mesh cell							
2								
3								
4								
.								
.								
IM-1								
IM								

The on-line printed output always includes a complete core memory storage map of the ODD-K program input data. The parameter, data, remark, or program routine label is printed with an octal number following the label. Two examples of the storage map printout are:

	<u>Array Label</u>	<u>Pieces of Data in Array</u>	<u>No. of Subscripts in Array</u>	<u>Initial Address</u>
Data Block:	A1	00003	1	17073
Parameter Block:	A00	00000	0	06610

In addition, if sense switch number 1 is down during the ODD-K problem, the on-line printout of the monitor line as described in the off-line printed data is obtained with each outer iteration.

4.2 TAPE OUTPUT

Binary tape output obtained from an ODD-K problem includes two FORTRAN IV binary tapes. These tapes are written if the user specifies this type of output for the problem.

The scalar flux data, $N0_{ig}$, is placed on a binary tape if $D01 = 2$ or 3 . This tape consists of a logical tape record for each group in the multigroup solution. Therefore, there are $GM(G01)$ records on the tape. The order of data within each group (logical record) is the mesh cell data for all radial mesh cells (IM) for the first axial row, the second axial row, to the JMth axial row. The tape is generated on the MSFC IBSYS version 13, tape number B-6. This tape must be saved if the data are to be used in a NAGS problem (Volume 7) or as input to the next ODD-K problem (e.g., the S_2 ODD-K neutron flux guess for an S_6 ODD-K problem).

The angular flux data, $B2_{ig}$, $B4_{ig}$, $B6_{ig}$, is placed on a binary tape if $L04 = 1$. This tape consists of a set of logical records for each group in the multigroup solution. The tape is generated on MSFC IBSYS version 13, tape number B-4. The actual data obtained from an ODD-K problem is nine logical records of which only three are actually needed for data processing in the DAFT program (Volume 8). The excess of data is obtained because the

numerical solution of the group fluxes are solved in two passes (i.e., downward and upward) through the mesh cell description, and the angular flux $B_{4,ig}$ and $B_{6,ig}$ is obtained each time the top $j = JM$, and bottom $j = 1$, mesh cell rows are passed. The fourth logical record of the set of nine for each group contains the angular flux at the bottom row. The seventh logical record contains the angular flux at the top row. The ninth logical record is the side surface (outer radius) angular flux, $B_{2,ig}$. The other records (first, second, third, fifth, sixth, and eighth) are excess data. This binary tape with nine records for each group, 1 to GM (G01), is edited by the DAFT code in the form generated by ODD-K.

5.0 PROGRAM OPERATION

The ODD-K program is loaded under the control of the FLOCOW II assembly program. (This program which is a self-contained monitor system is explained in some detail in Appendix A.) The FLOCOW assembly program is similar to its predecessor, the FLOCO II assembly program, except FLOCOW is compatible with the independent FORTRAN II monitor system of the IBM 7094 computer. The input and output data from a program run under FLOCOW uses the normal input and output channels of the FORTRAN II monitor system; the binary tape operations of FLOCOW are normal FORTRAN II or FORTRAN IV binary tapes.

The ODD-K program is separated into two sub programs. The first sub program's function is, to read all parameter data and all large group dependent arrays of data. The group dependent data considered to be large are: the cross sections (C0), fluxes (N0), and fixed distributed sources (Q0). These data are processed by the first program and loaded onto separate binary tapes as a logical record for each group of data. These tapes are then used by the next segment of the ODD-K program. The second program of ODD-K which is loaded by FLOCOW after the Q0 or N0 data, performs the bulk of the numerical solution in ODD-K. This solution is carried out as a single group solution in core memory (i.e., only one group of C0, N0, or Q0 data is in core at any one time), and data for all other groups are on the binary work tapes.

5.1 TAPE ASSIGNMENTS

The tape assignments of ODD-K in the MSFC IBSYS version 13 system are shown in Table 3 with the function of each tape noted. In using ODD-K, the user must specify tape mountings and dismountings to be done at the beginning and completion of the ODD-K problem.

5.2 SENSE SWITCH OPERATIONS

Sense switch operations are performed either to print the monitor line on-line or force a final print of data. The operations have no effect if the sense switch is UP. The sense switch DOWN operations are:

Sense Switch 1: Print monitor line on-line

Sense Switch 4: Force pseudo-convergence and final off-line print

The sense switch DOWN operation occurs only at the completion of an outer iteration. The 7094 operator must be told to wait until a time elapse of one outer iteration (up to 15 minutes for a large job) if a final print is desired.

Table 3
ODD-K TAPE ASSIGNMENTS IN THE MSFC IBSYS VERSION 13

<u>Tape No.</u>	<u>System Function</u>
A-1	System tape
A-2	BCD input tape
A-3	Binary input tape (group fluxes)
A-4	Binary input tape (fixed distributed sources)
A-5	Intermediate work tape (group cross sections)
A-6	Binary input tape (master cross section tape)
A-7	Not used
A-8	Not used
A-9	Intermediate work tape (group fluxes and currents)
A-10	Not used
B-1	BCD output tape
B-2	BCD punch output tape
B-3	Intermediate work tape (group fixed distributed sources)
B-4	Binary output tape (group angular fluxes)
B-5	Not used
B-6	Binary output tape (group scalar fluxes)
B-7	Not used
B-8	Not used
B-9	Intermediate work tape (group fluxes and currents)
B-10	Not used

5.3 PROGRAM STOPS

There are two types of program stops that can occur. These are: 1) data error stops, and 2) tape redundancy stops. The data errors will be checked and noted on the on-line printout by FLOCOW. If an error is detected, execution of the problem will be terminated, a remark will be printed on-line, and FLOCOW will proceed to the next job card and load the FORTRAN monitor system. The standard data errors and their possible meanings are:

Code Block: 804, Angular quadrature constants in error. Input error, check M7, M5, and W0.

810, Radial or axial mesh coordinate dimensions in order $R0_{IM+1} = 0$, $R0_{i+1}$. Input error.

811, Geometry constants in error $V1_g = 0$. Input error.

813, Parametric eigenvalue $S03 = 0$ when $S02 \neq 0$. Input error.

814, $S04 \neq 0$. Check Q0 input data.

816, $V7_g = 0$. Input error.

822, Fission totals or $\Lambda = 0$. Check Q0, F0, I0, I1, I2 data blocks.

823, Check K7, V7, F0, V0, R0 blocks.

834, Check I02, M2 block. Cross sections in error.

841, Check Q0 block or cross section table specifications HT, HS, HL.

851, Check K7 block, $K7 \neq 1.0$.

852, $I06 = 0$ for problems other than Q or K. Two successive eigenvalues equal.

The tape redundancy errors are printed on-line. If the operator is instructed to follow the on-line print, the tape error can be corrected and the ODD-K job restarted. If the job is not restarted, the problem must be resubmitted. The tape error is noted on the 7094 console as a HPR 700 in the storage register.

5.4 ASSEMBLY ERROR STOPS

It is possible for the assembly program to find some errors that may be in the input deck. If those errors are detected, the loading stops and the 7094 storage register shows an HPR instruction with an octal address from zero to seven, indicating which type of error has occurred

The operator should note this stop on the job card, and try to give an absolute core memory dump. The eight types of error stops which are noted by FLOCOW as the address of the HPR instruction are:

<u>Address</u>	<u>Meaning</u>
0	Information error, check deck setup
1	Illegal operation, ODD-K program deck is bad or is loaded wrong
2	Address symbol undefined, ODD-K program deck is bad or is loaded wrong
3	Check sum error, FLOCOW program deck is bad
4	ODD-K program deck is bad
5	Memory exceeded, ODD-K problem size is too large
6	Tape redundancy error
10	End of tape encountered on BCD output tape B-1. Change tape and proceed

Stop #0 occurs, for instance, if a data block is being loaded and an I or R card is encountered. Generally, this stop is an indication that the cards are mixed up or that the control punch is wrong.

Stop #1 takes place, for example, if a Type B operation code, not in the tables on pages 26 - 27 of Reference 3 is specified; however, the stop may occur for other reasons. The operation code may have been mispunched or placed in the wrong columns. A 9 or R card may be among the program deck (I cards) cards.

Stop #2 occurs if a symbol is encountered, generally in the address of the instruction, which has not been previously defined, i. e., the control word is found to be zero which means that no space has been assigned to the symbol. This stop often indicates that the ODD-K program deck is not arranged properly.

Stop #3 is that the loading of the FLOCOW binary cards has met with a check sum failure.

Stop #4 is that Type A instructions must have a non-zero tag and violations lead to loading stops.

Stop #5 is that the next block address (NBA), which is calculated during the loading and assembly of the ODD-K program and data, exceeds the available core memory. If NBA

exceeds the core memory capacity of the IBM 7094 computer (32,718 locations), the loading stops and this error is noted.

Stop #6 the redundancy tape test has failed three times, and the problem should either be restarted or resubmitted.

Stop #10 is that an end of BCD output tape B-1 has been encountered. Change to a new tape and press start button to resume printing.

SECTION

6.0 REFERENCES

1. Worlton, J. W., and Lathrop, K. D., Los Alamos Scientific Laboratory, Personal Communication.
2. Carlson, B. G., and Lathrop, K. D., "Transport Theory, The Method of Discrete Ordinates",
3. Lathrop, K. D., and Carlson, B. G., "Discrete Ordinates Angular Quadrature of the Neutron Transport Equation", LA-3186, February 12, 1965.
4. Carlson, G. G., et. al., "FLOCO-II Manual", LAMS-2339, October 13, 1959.
5. Putnam, G. E., "TOPIC, A FORTRAN Program for Calculating Transport of Particles in Cylinders", IDO-16968, April, 1964.
6. Putnam, G. E., and Shapiro, D. M., "MIST, Multigroup Internuclear Slab Transport", IDO-16856, Internuc-67, May, 1963.

APPENDIX A

FLOCOW ASSEMBLY PROGRAM

This appendix is included to describe the FLOCOW assembly program. The following discussion is not intended for the ODD-K program user, but rather for a system programmer proficient in FAP machine language operations.

The FLOCOW assembly program is a modified and improved version of Los Alamos FLOCO II assembly program and may be used as a direct replacement for FLOCO II.

The major differences between FLOCOW II and FLOCO II are:

- 1) The FLOCOW II is designed for use with tape-oriented IBM 7090/94 with off-line input and output, whereas FLOCO II is essentially an assembly program for on-line programs.
- 2) FLOCOW II permits direct communication with FORTRAN programs via binary tapes.
- 3) FLOCOW II requires approximately 1000_{10} fewer memory core storage locations than the later versions of FLOCO II.

To avoid misinterpretation of the terminology used in this report, the term FLOCO used alone refers to the FLOCO programming system; the assembly programs for this system are always referred to as FLOCOW or FLOCO II.

This report is not a complete FLOCO system manual, but rather updates and enlarges upon the basic FLOCO II Manual as reported in Reference 3.

OPERATION UNDER MONITOR CONTROL

FLOCOW currently operates within the confines of the independent IBM FORTRAN II Monitor System (non-IBSYS FORTRAN II), but is not under direct system control. The FLOCOW deck and the FLOCO program, to be executed, must be loaded onto a monitor system input tape (A2). Upon execution of the FLOCO job, the monitor system will cause FLOCOW to be loaded and then program control is passed to the FLOCOW assembler which loads and assembles the program to be executed.

After completion of the FLOCO program, a normal exit returns control to the FORTRAN II monitor system for processing of the next job.

Off-line print appears on the monitor system output tape (B1) and binary cards are punched from the system punch tape (B4).

Operation under the MSFC IBSYS Version 13 system is accomplished as a SPOOK FORTRAN job run as a non-system job. At present, the only monitor control cards used are *FAP, *XEQ, *STOP, and *DATA. The deck order is: cards, *FAP, *XEQ, *STOP, FLOCOW DECK, *DATA, Program and Problem Data Decks.

MACHINE REQUIREMENTS

The FLOCOW assembly system requires a minimum 7090 but will assemble FLOCO programs containing some 7094 instructions. The standard FLOCO II instruction set as described in Reference 3 is used, and FLOCOW does not contain the following IBM 7094 instructions:

- A) any instruction referring to the indicators
- B) any instruction referring to channels C through H
- C) any convert instruction
- D) any double precision instructions
- E) any hypertape instructions
- F) the following miscellaneous instructions: ENB, ESNT, VSM, VDH, VDP, VIM, and SDN.

If it becomes apparent that any of these instructions are needed, it is a simple matter to add to the FLOCOW dictionary. The following instructions should be entered as PSE or MSE instructions:

BTT - PSE	xxxx	ECTM - MSE	6
FRN - PSE	11	EMTM - MSE	16
IOY - PSE	5	ESTM - MSE	5
LMTM - PSE	16	ETT - MSE	xxxx
RCT - PSE	15	LSNM - MSE	10
RDCE - PSE	1352		
RDCB - PSE	2352		

Since FLOCOW accepts only three-letter operation codes, the following instructions use mnemonic codes different from the SHARE standard codes:

EFTM - EFM	TCNA - TNA
LFTM - LFM	TCNB - TNB
RCHA - RCA	TEFA - TEA
RCHE - RCB	TEFB - TEB
SCHA - SHA	TRCA - TCA
TCOA - TOA	TRCB - TCB
TCOB - TOB	

MULTIPLE TAG MODE

FLOCOW programs are normally run in the multiple tag mode. If it is desired to use seven index registers, the programmer must execute a leave multiple tag mode (PSE 16) instruction.

Before entering any of the built-in input/output routines, the program must be in the multiple tag mode. The exit routine is exempt from this requirement.

BUILT-IN ROUTINES

The following named routines are available in FLOCOW:

FLOCOW Fortran Binary Tape Program - Routine 966

"	Exit routine	"	967
"	Loading program	"	970
"	Data Assign program	"	971
"	Convert Symbolic program	"	972
"	Punch Binary program	"	973
"	Print Off-Line program	"	974
"	Print On-Line program	"	975
"	Print storage Map program	"	976
"	Read/Write Tape program	"	977

Descriptions of those routines, adequate for programming purposes, are described below and system programmers should consult these descriptions and Reference 3 for details.

As with FLOCO II, entry to all these routines is via a TSX49xx instruction; the TSX49xx is followed by the appropriate calling sequence. The prefix of each call is zero except for the last call in the sequence, whose prefix is the numeral 4. For example, the

sequence of instructions

TSX4973		TSX4973
00000Z5	OR	0 Z5
40000N3		4 N3

will cause the data sets labeled Z5 and N3 to be punched on column binary cards. The prefix must not be omitted or the sequence will result in error stop.

Except for routines 967 and 970, return from these routines is to the first instruction following the calling sequence.

MSFC TAPE ASSIGNMENTS

IBSYS Version 13 Physical Tape	FORTTRAN IV* Logical Tape Number	FLOCO Tape Number	Physical Tape	FORTTRAN Logical Tape No.	FLOCO Tape Number
A1	System	1	B1	6	5
A2	5	12	B2	7	6
A3	1	11	B3	2	3
A4	3	4	B4	4	13
A5	8	2	B5	9	14
A6	10	7	B6	11	10

Tape A1 A1 is the IBSYS system tape and should never be addressed by a FLOCO program.

Tapes A2, B1, B2 A2, B1, and B2 are MSFC IBSYS system input/output tapes and are addressed via the appropriate FLOCOW input/output routine. Only a skilled programmer should attempt to address them directly.

Tapes A4, B3, B4 A4, B3, and B4 are the FORTRAN system standard utility tapes; since tapes are usually mounted on these drives, they should be used for the same purpose in FLOCO programming. They may be addressed directly or via routines 966 or 977.

Tapes A5, A6, B5, B6 A5, A6, B5, and B6 are the FORTRAN system non-standard utility tapes and, thus should be used whenever it is desirable to save a tape. It is suggested that, before terminating any job, the tape to be saved should

*FORTRAN IV tapes are illustrated since ODD-K output is used in FORTRAN IV programs.

be rewound and unloaded. These tapes may be addressed either directly or via routines 966 or 977.

ROUTINE 966: FORTRAN BINARY TAPE PROGRAM

This routine is used, either to write a binary tape which may be read by a FORTRAN program, or to read the binary output of a FORTRAN program. (It uses routine 977 for the actual input/output.) It has two types of entries in its calling sequence. A type "A" call is used to set up the proper block size and to select the desired tape unit and tape density. A type "B" call is used to cause the desired data to be read or written. The calling format is

	<u>Prefix</u> - <u>Decrement</u> - <u>Tag</u> - <u>Address</u>
Type A Call:	P - Block Size - Density - Tape No.
Type B Call:	P - Operating System - read/write tag - Sx/8xx
where:	
P = 0, 4	0 - except for last call of sequence 4 on last call of any sequence
"Block Size"	is the maximum number of words per physical record. It is normally 200 (128 decimal) with FORTRAN II programs, and 400 with FORTRAN IV.
"Density" = 0, 1	0 - tape unit is written or read in low density 1 - tape unit is written or read in high density
"Tape No. "	The FLOCO tape number of the tape to be selected
"Operating System"	0 or blank - FORTRAN II under Independent FORTRAN Monitor System 1 - FORTRAN II or IV under IBSYS
"Read/Write Tag" = 1, 2	1 - Read from specified tape 2 - Write on specified tape
"Sx/8xx/xxx"	The symbolic name of the data (S(x)), subroutine (8xx), or FLOCO relative address of any material to be written or read.

As an example, the calling sequence

```
REW 7  
TSX4966  
02001007  
00001Z5  
40001N0  
RUN 7
```

rewinds tape A6, and then reads two logical records from that tape in 128 word high-density blocks. The data on tape A6 was presumably generated by a FORTRAN II program operating under the FORTRAN Monitor System. The first record is read into the space reserved for the data labeled Z5. The second record into N0. The program then rewinds and unloads A6.

The one type "A" call can serve several type "B" calls, as long as the information carried by the type "A" call does not change and if no built-in I/O routine other than 966 is entered between the type "A" and its following type "B" calls.

Tape Format and FORTRAN Formatting

Internal matrix storage under FLOCOW is identical to that used in FORTRAN II. Routine 966 reads and writes tapes as if they were produced by an unformatted FORTRAN statement. Thus, the following FORTRAN II reading routines will yield a matrix stored in the normal fashion:

```
Dimension: A (IMAX, JMAX)
```

```
READ TAPE 9A
```

The remaining paragraphs of this section contain details not necessary for the casual use of FLOCOW.

FLOCO "formulae" are stored sequentially in increasingly higher core locations; one-dimensional data can be transmitted without reversal by means of a type "B" call with 8xx address. Here 8xx is either FLOCO Formula 8xx or a "Relative Address" in FLOCO form Xxx. In either case, the decrement of the location 8xx (or Xxx) must contain the number of data words to be transmitted and the address must contain the location of the first (lowest) data

word to be transmitted. For example, consider the following coding sequences:

TSX 4966
0 200 1005
4 2803

TSX 4966
0 200 1003
4 2x10

TSX 4966
0 200 1003
4 2x10

(X10) 1 A01 0X11

(X10) 1 A01 0803

The first sequence results in Formula 803 being written on Tape B1. The number of words written are determined by the decrement of location 803. This decrement set either by a "load instruction" pseudo-operation (80000803), or by the programmer. (Never attempt any FLOCO controlled I/O of a formula defined solely by a "Record Origin" pseudo-operation (80000803), for then location 803 has zero decrement.) The second example transmits A01 words of data starting with location X11. The third example transmits the first A01 words in formula 803. The prefix of instruction X10 in the third case must not be 0 or 4 because of the peculiarities of the FLOCOW "C-type" instruction (see pages 13 and 25 of Reference 4.).

CAUTIONS

CAUTION 1: Routine 966 makes no assumption about tape positioning before reading or writing except to check for beginning and end of tape. Tape positioning may be accomplished directly or through Routine 977.

CAUTION 2: Reading takes place in the binary mode. Any error in mode or density results in FLOCO error stop 6.

CAUTION 3: It is unwise to attempt to read records of the wrong size with Routine 966, but an excessive number of words can never be read. Thus, it is impossible to erroneously overlay other data by reading oversize records. Attempting to read an improperly dimensioned record generated by FORTRAN operating under IBSYS results in FLOCO error stop 7. The same stop may occur when improperly reading a tape generated under the Independent FORTRAN Monitor System.

CAUTION 4: Tape errors are handled differently in different FORTRAN versions; although difficulty has not been experienced, Routine 966 does not handle all possible bad tape conditions satisfactorily. It is expected that the most frequent trouble will involve FORTRAN II tapes generated by Routine 966 which are unacceptable to the Independent Monitor System.

Incompatibility with tapes generated by IOEX under IBSYS should be rare. Such tapes may be corrected by reading them under the system used in their generation and followed by rewriting on a good tape.

ROUTINE 967: EXIT ROUTINE

Entry into this routine returns control to the monitor system which then proceeds to the next job. Note: as with all references to FLOCO subroutines, the transfer instruction to enter 967 must be tagged, i.e., TSX 4967 will be executed properly, but the instruction TSX 967 or TRA 967 will result in an assembly error and FLOCO stop 4.

ROUTINE 970: LOADING ROUTINE

This routine is similar to the FLOCO II routine 970 except that loading takes place from tape A2 rather than the card reader. Share relocatable binary cards are relocated from 230_8 as was done in the older versions of FLOCO II.

This routine recognizes comment cards (* in col. 1) and prints them both on- and off-line. Once control has passed to FLOCOW, monitor control cards with an asterisk in column 1 (e.g., *STOP) are treated as commentary.

Return from this routine may be through the usual FLOCO transfer card I* xxx48xx which transfers to the $(xxx)_{8th}$ instruction of routine 8xx.

In addition, FLOCOW recognizes a new type of input card, the FLOCOW CONTROL CARD, which is any card with column 1 blank. Upon reading this card, FLOCOW transfers to routine 960. Routine 960 is supplied by the programmer. Further control information may be included on the control card as the contents of cols. 1-72 and stored in the addresses of locations $230_8 - 337^*_8$.

*As right justified octal integers or B.C.D. hollerith - except that zero and blank are interchanged, i.e., the word A10 638 would be stored in six consecutive locations as $21_8, 01_8, 60_8, 0_8, 3_8$

While error exits may be made directly through routine 967, it is best to return to routine 970 upon the normal completion of a problem. This permits the reading of stacked problems and gives a proper return to the monitor system. Normally an End of File (tape mark) is placed between jobs on the system input tape. Upon reading this tape mark, routine 970 transfers to the exit routine 967, thus leaving the input tape properly positioned.

Input tape errors are sensed, appropriate comments printed, and improper action taken.

FLOCOW routine 970 reads all card formats read by FLOCO II. Since the over-punched zero (11-0, 12-0) is not a legal Hollerith symbol, but is acceptable in the FLOCO system, the interpretation of this character depends on the card-to-tape equipment used to prepare the input tape. FLOCOW assumes $\overset{+}{0}$ will enter the core as 32 octal and $\bar{0}$ as 52 octal. All other acceptable FLOCO symbols will be correctly interpreted.

ROUTINES 971 AND 972

See Reference 4.

ROUTINE 973

The only difference between FLOCOW routine 973 and FLOCO II routine 973 is that the FLOCOW routine places binary card images on the system punch tape (B4). These card images are identical to FLOCO II binary card images except that they contain a 7-9 punch in column 1. As a result, the off-line tape-to-card equipment will punch column binary cards rather than row binary cards. FLOCO system binary output has a 1 punch in the tag of the first word of each card (6 punch in col. 2). Since the column is ignored by the BSS loader, these cards can be used as input for FORTRAN programs.

ROUTINES 974 AND 975: PRINT ROUTINES

The FLOCOW versions are nearly identical to the FLOCO II routines with off-line printing placed on tape A3. The major difference is that remark cards are printed over 120 columns instead of 72 columns. This permits each column of output to be properly headed. The

first card of a "remark" routine contains column 1-72 of the first line, the second card contains columns 74-120 and 2-24 of the second line, etc. The first column of every line of printed output is blank and so are the printed columns corresponding to the first column of the second and succeeding remark cards.

ROUTINE 976

See Reference 4.

ROUTINE 977

See Reference 4. It should be pointed out that this routine results in the reading and writing of low density binary records of arbitrary length. The address of type B calls for routine 977 may be of Xxx type discussed under routine 966.

Finally, in the past, it has been necessary to use two successive REW or RUN instructions before entering this routine. This is no longer necessary; the execution of the guilty RDC instruction has been properly modified.