

CONTENTS

ABSTRACT.....	vii
I. INTRODUCTION.....	I-1
II. OVERVIEW OF THE ONEDANT CODE PACKAGE.....	II-1
A. Programming Practices and Standards.....	II-1
1. Language.....	II-1
2. Structure.....	II-1
3. Standard Interface Files.....	II-1
4. Data Management and Transfers.....	II-2
5. Central Memory Restrictions.....	II-2
6. Word Size.....	II-2
B. ONEDANT Code Package Structure.....	II-2
C. INPUT Module.....	II-6
D. SOLVER Module.....	II-7
E. EDIT Module.....	II-12
III. CARD-IMAGE INPUT FORMAT RULES.....	III-1
A. Free-Field Input.....	III-1
1. Card-Image Ground Rules.....	III-1
2. Delimiters (Separators) and Terminators.....	III-1
3. Numerical Data Item Ground Rules.....	III-1
4. Hollerith Data Item Ground Rules.....	III-2
5. Array Identification and Ordering.....	III-2
6. Block Identification and Ordering.....	III-2
7. Input Data Operators.....	III-2
B. User-Specified Input Formats.....	III-3
C. Fixed-Field FIDO Input.....	III-6
1. Card-Image Ground Rules.....	III-6
2. Delimiters (Separators) and Terminators.....	III-6
3. Numerical Data Ground Rules.....	III-6
4. Hollerith Data Item Ground Rules.....	III-6
5. Array Identification and Ordering.....	III-7
6. Block Identification and Ordering.....	III-7
7. Input Data Operators.....	III-7
IV. ONEDANT CODE PACKAGE INPUT SPECIFICATIONS.....	IV-1
A. Overview of the Specification of Input.....	IV-1
B. ONEDANT INPUT: Mini-Specification Sheet.....	IV-5
C. Input Specifications.....	IV-7
V. DETAILS RELATED TO INPUT FOR ONEDANT.....	V-1
A. Development of the Multigroup, Discrete-Ordinates Form of the Transport Equation.....	V-1
1. Particular Forms of the Divergence Operator.....	V-3
2. Spherical Harmonics Expansion of the Scattering Source.....	V-4
3. Spherical Harmonics Expansion of the Inhomogeneous Source.....	V-6
4. Discretization of the Energy Variable-the Multigroup Approximation.....	V-9
5. Discrete-Ordinates Equations.....	V-10
a. Standard Plane Geometry.....	V-11
b. Two-Angle Plane Geometry.....	V-13
c. Cylindrical Geometry.....	V-14

d.	Spherical Geometry.....	V-15
e.	Starting Directions.....	V-16
6.	Discretization of the Spatial Variable.....	V-17
B.	Input of Geometry-Related Information.....	V-17
C.	Input of the Basic Cross-Section Library.....	V-18
1.	ISOTXS and GRUPXS Standard Interface Files.....	V-18
2.	Card-Image, BCD Libraries.....	V-19
a.	Ordering of Cross Sections within a Cross-Section Table..	V-19
b.	Card-Image Data Formats.....	V-19
c.	Cross-Section Table Title Cards.....	V-20
d.	Anisotropic Scattering and the Ordering of Cross-Section Tables.....	V-20
3.	Binary Form of Card-Image Libraries.....	V-22
4.	MACRXS and SNXEDT Cross-Section Files.....	V-22
5.	Coupled Neutron-Gamma Cross Section Sets.....	V-25
D.	Material Mixing and the Creation/Use of Interface Files.....	V-25
1.	Material Mixing and the Creation of Interface Files.....	V-25
2.	Using Existing MACRXS, SNXEDT, NDXSRF, ZNATDN Interface Files.....	V-27
E.	Assignment of Materials to Zones.....	V-28
F.	Input of Inhomogeneous Sources.....	V-29
1.	Distributed Source Input.....	V-29
2.	Surface (Boundary) Source Input.....	V-31
G.	Input of Quadrature Sets.....	V-32
H.	Boundary Conditions.....	V-33
I.	Buckling Corrections.....	V-34
J.	Eigenvalue Searches.....	V-35
K.	Adjoint Computations.....	V-39
L.	The MAXLCM Parameter.....	V-40
VI.	DETAILS RELATED TO SOLVER MODULE EXECUTION.....	VI-1
A.	Iteration Strategy.....	VI-1
B.	Convergence Criteria.....	VI-4
1.	Inner Iteration Convergence.....	VI-4
2.	Diffusion Sub-Outer Iteration Convergence.....	VI-5
3.	Full Convergence.....	VI-5
4.	Iterative Loop Termination.....	VI-6
C.	Iteration Monitor Print.....	VI-7
VII.	DETAILS RELATED TO EDIT MODULE INPUT AND EXECUTION.....	VII-1
A.	Spatial Options for Edits.....	VII-1
B.	Energy-Group Options for Edits.....	VII-3
C.	Forms of Response Functions.....	VII-4
1.	Cross-section Response Functions: EDXS Input Array.....	VII-4
a.	Resident Macroscopic Cross-Section Response Functions: RESDNT Input Parameter.....	VII-5
b.	Isotope Microscopic Cross-Section Response Functions: EDISOS Input Array.....	VII-5
c.	Resident Constituent Cross-Section Response Functions: EDCONS Input Array.....	VII-6
d.	Material Cross-Section Response Functions: EDMATS Input Array.....	VII-6
2.	User-Input Response Functions: The RSFE and RSFX Input Arrays.....	VII-6

D.	Response Function Summing Options.....	VII-7
1.	Cross-Section Response Functions Sums: MICSUM Input Array.....	VII-7
2.	User-Input Response Function Sums: IRSUMS Input Array.....	VII-8
E.	Adjoint Edits.....	VII-9
VIII.	CONTROLLING THE EXECUTION OF MODULES AND SUBMODULES.....	VIII-1
A.	Module Execution Control.....	VIII-2
1.	INPUT Module Execution Control.....	VIII-2
2.	SOLVER Module Execution Control.....	VIII-2
3.	EDIT Module Execution Control.....	VIII-2
B.	INPUT Submodule Execution Control (File Generation Suppression).....	VIII-3
1.	Geometry Submodule Execution Control.....	VIII-3
2.	Mixing Submodule Execution Control.....	VIII-3
3.	Assignment-of-Materials-to-Zones Submodule Execution Control.....	VIII-4
4.	Working-Cross-Section-File Submodule Execution Control.....	VIII-4
5.	SOLVER-Input-File Submodule Execution Control.....	VIII-5
6.	EDIT-Input-File Submodule Execution Control.....	VIII-5
7.	Adjoint-Reversal Submodule Execution Control.....	VIII-5
IX.	ERROR DIAGNOSTICS.....	IX-1
A.	Examples of Errors and Resulting Messages.....	IX-1
B.	Comments Regarding Multiple Errors.....	IX-7
APPENDIX A.	Code-Dependent Interface File Descriptions.....	A-1
APPENDIX B.	ONEDANT Sample Problems.....	B-1
REFERENCES.	R-1

USER'S MANUAL FOR ONEDANT:

A CODE PACKAGE FOR ONE-DIMENSIONAL, DIFFUSION-ACCELERATED,
NEUTRAL-PARTICLE TRANSPORT

by

R. Douglas O'Dell, Forrest W. Brinkley, Jr., and Duane R. Marr

ABSTRACT

1. Program Identification: ONEDANT
2. Computer for which Program is Designed: CDC-7600, but the program has been implemented and run on the IBM-370/190 and CRAY-I computers.
3. Function: ONEDANT solves the one-dimensional multigroup transport equation in plane, cylindrical, spherical, and two-angle plane geometries. Both regular and adjoint, inhomogeneous and homogeneous (k_{eff} and eigenvalue search) problems subject to vacuum, reflective, periodic, white, albedo, or inhomogeneous boundary flux conditions are solved. General anisotropic scattering is allowed and anisotropic inhomogeneous sources are permitted.
4. Method of Solution: ONEDANT numerically solves the one-dimensional, multi-group form of the neutral-particle, steady-state form of the Boltzmann transport equation. The discrete-ordinates approximation is used for treating the angular variation of the particle distribution and the diamond-difference scheme is used for phase space discretization. Negative fluxes are eliminated by a local set-to-zero-and-correct algorithm. A standard inner (within-group) iteration, outer (energy-group-dependent source) iteration technique is used. Both inner and outer iterations are accelerated using the diffusion synthetic acceleration method.
5. Restrictions: The code is thoroughly variably dimensioned with a flexible, sophisticated data management and transfer capability. The code is designed for a three-level hierarchy of data storage: a small, fast core central memory (SCM), a fast-access, peripheral large core memory (LCM), and random-access peripheral storage. (For computing systems based on a two-level hierarchy of data storage - a large fast core and random-access peripheral storage - a portion of fast core is designated as a simulated LCM to mimic the three-level hierarchy). Random-access storage is used only if LCM (or simulated LCM) storage requirements are exceeded. Normally, an SCM of about 25 000 words of storage and an LCM (or simulated LCM) of a few hundred thousand words or less storage are sufficient to eliminate the need for using random-access storage.

6. **Running Time:** Running time is directly related to problem size and to central processor and data transfer speeds. On the CDC-7600 a 70 energy-group, S_{16} , P_3 scatter, 40 space-point eigenvalue problem requires about 40 sec CPU time.

A 42 energy-group, S_8 , P_3 scatter, 121 space-point fixed-source problem requires about 30 sec CPU time on the CDC-7600.

A 1 energy-group, S_{48} , P_0 scatter, 307 space-point fixed-source problem requires from 2 to 3 sec CPU time on the CDC-7600.

Generally, then, on the CDC-7600, the running times for ONEDANT will range from a few seconds to 1 or 2 minutes.
 7. **Unusual Features of the Program:** The ONEDANT code package is modularly structured in a form that separates the input and the output (or edit) functions from the main calculational (or solver) section of the code. The code makes use of binary, sequential data files, called interface files, to transmit data between modules and submodules. Standard interface files whose specifications have been defined by the Reactor Physics Committee on Computer Code Coordination are accepted, used, and created by the code. A totally new free-field card-image input capability is provided for the user. The code provides the user with considerable flexibility in using both card-image or sequential file input and also in controlling the execution of both modules and submodules. Separate versions of the code exist for short-word and long-word computers.
 8. **Programming Languages:** The program is written in standard FORTRAN-IV language.
 9. **Machine Requirements:** Six Input/Output units and up to 14 interface units are required. The number of interface units needed is problem dependent. Typically, 10 such units are used. For CDC-7600 computers a 50 000 word small core (SCM) and large core memory (LCM) are required. For computers with only a single fast core, the fast core size must be sufficiently large to permit partitioning into an SCM and simulated LCM. Random-access auxiliary storage may occasionally be required if LCM (or simulated LCM) storage is insufficient for the problem being executed.
 10. **Material Available:** Source deck (about 30 000 card-images), sample problems and this manual have been submitted to the Argonne Code Center and to the Radiation Shielding Information Center.
-

I. INTRODUCTION

The ONEDANT code package is a modular computer program designed to solve the one-dimensional, time-independent, multigroup discrete-ordinates form of the Boltzmann transport equation. The modular construction of the code package separates the input processing, the transport equation solving, and the post-processing, or edit, functions into distinct, independently executable code modules, the INPUT, SOLVER, and EDIT modules, respectively. These modules are connected to one another solely by means of binary interface files. The INPUT module and, to a lesser degree, the EDIT module are general in nature and are designed to be standardized modules. With these modules, different new production codes can be created simply by developing different SOLVER modules that can be "plugged in" to the standardized INPUT and EDIT modules. ONEDANT, employing a one-dimensional, time-independent transport equation SOLVER module, is the first production code to be issued by Los Alamos using this modular construction. Some of the major features and improvements included in the ONEDANT package are:

- (1) a totally new, free-field format card-image input capability designed with the user in mind,
- (2) highly sophisticated, standardized, data- and file-management techniques as defined and developed by the Committee on Computer Coordination (CCCC) and described in Ref. 1; both sequential file and random-access file handling techniques are used,
- (3) the use of a diffusion synthetic acceleration scheme to accelerate the iterative process in the SOLVER module,
- (4) direct (forward) or adjoint calculational capability,
- (5) standard plane, two-angle plane, cylindrical or spherical geometry options,
- (6) arbitrary anisotropic scattering order,
- (7) vacuum, reflective, periodic, white, albedo, or surface source boundary condition options,
- (8) inhomogeneous (fixed) source or k_{eff} calculation options as well as time-absorption (alpha), nuclide concentration, or dimensional search options,
- (9) "diamond-differencing" for solution of the transport equation,

- (10) user flexibility in using both card-image or sequential file input,
- (11) user flexibility in controlling the execution of both modules and submodules, and
- (12) extensive, user-oriented error diagnostics.

ONEDANT is a large, very flexible code package. Great effort has been devoted to making the code highly user-oriented. Simple problems can be easily run and many of the code options can be ignored by the casual user. At the same time numerous options for selective and sophisticated executions are available to the more advanced user. In all cases redundancy of input has been minimized, and default values for many input parameters are provided. The code is designed to be "intelligent" and to do much of the work for the user. The input is designed to be meaningful, easily understood, easily verified, and easy to change. The printed output is well documented with liberal use of descriptive comments and headings. In short, ONEDANT was designed to be fun to use.

Chapter II of this manual provides the user with an overview of the code package. Included are sections on programming practices and standards, code package structure, and functional descriptions of the three principal modules comprising the package.

Chapter III presents the card-image input format rules for the user.

Chapter IV provides the card-image input specifications for ONEDANT. First is given an overview of the specification of input including descriptive examples. Next is a "mini-specification" sheet on which are listed all the available input arrays arranged by input block. This sheet is very useful to the user in organizing his input. For the more experienced user, the mini-specification sheet is frequently all that is needed for him to specify his input. Following the mini-specification sheet is a moderately detailed description of all the input parameters and arrays.

Chapters II, III, and IV should be read by all first-time users of ONEDANT.

Chapter V provides the interested user with details related to the input for ONEDANT. Included is a brief development of the multigroup, discrete-ordinates form of the diamond-differenced Boltzmann transport equation. This section is followed by numerous sections providing specific detailed information needed by the user to fully understand some of the input options and input arrays. The chapter supplements the information presented in Ch. IV.

Chapter VI gives details related to the actual execution of the SOLVER module. Described are the iteration strategy, convergence criteria, termination criteria for the iterative loops, and the iteration monitor print provided by the code.

Chapter VII is devoted to details related to the EDIT module of the code. Both input and execution-control options for this module are described in detail. This chapter supplements information pertaining to the EDIT module provided in Ch. IV.

In Chapter VIII is a discussion of some of the more sophisticated options available to the advanced user for controlling the execution of modules and submodules in ONEDANT.

Chapter IX presents a discussion of the error diagnostics available in ONEDANT. Several examples of errors and the resulting error messages are provided for the user.

Two appendices are also included in the manual. Appendix A provides the file descriptions for the code-dependent, binary, sequential interface files generated by and used in the ONEDANT code package. File descriptions for the CCCC standard interface files are not provided, but can be found in Ref. 1. Appendix B provides several sample problems for the user.

II. OVERVIEW OF THE ONEDANT CODE PACKAGE

The ONEDANT code package is a computer program designed to solve the one-dimensional, multigroup, discrete-ordinates form of the neutral-particle Boltzmann transport equation. It was developed as a modular code package consisting of three modules: an INPUT module, a SOLVER module, and EDIT module.

In this chapter is provided a discussion of the general programming practices and standards used in the code package, a description of the code structure, and overviews of the three modules comprising the package.

A. Programming Practices and Standards

In general the programming standards and practices recommended by the Committee on Computer Code Coordination (CCCC)^{1,2} have been followed throughout the development of ONEDANT. By following these practices and standards, problems associated with exporting and implementing the code in different computing environments and at different computing installations are minimized. This section provides a brief summary of the CCCC programming practices and standards used in ONEDANT.

1. Language. The programming language is standard FORTRAN as defined by the ANSI standard X3.9-1966.³

2. Structure. The code is structured in a form that separates the input and the output(or edit) functions from the main calculational (or solver) section of the code. A more complete description of the code structure is provided in Section B of this chapter.

3. Standard Interface Files. ONEDANT makes use of interface files to transmit data between and within its modules. These interface files are binary, sequential data files. Standard interface files are interface files whose structure and data-content formats have been standardized by the CCCC. Code-dependent interface files are files whose structure and data-content formats have not been standardized.

The following CCCC standard interface files are accepted, created, or otherwise used in ONEDANT: ISOTXS, GRUPXS, GEODST, NDXSRF, ZNATDN, SNCONS, FIXSRC, RTFLUX, ATFLUX, RAFLUX, and AAFLUX. File descriptions for these files are provided in Ref. 3.

The following code-dependent interface files are used in ONEDANT: MACRXS, SNXEDT, ADJMAC, SOLINP, EDITIT, and ASGMAT. File descriptions for these files are provided in Appendix A.

The use of the above interface files is described in Section B of this chapter.

4. Data Management and Transfers. ONEDANT is designed with rather sophisticated data-management techniques in order to accommodate, as efficiently as possible, the transfer of the large amounts of data frequently needed for solving large problems. Data management in the code involves the reading and writing of sequential data files, a flexible capability to block data, and if needed, use of multilevel data-management/transfers using random-access files.

The CCC standard subroutines SEEK, REED, and RITE are used for data transfers involving binary, sequential data files. A description of these routines is provided in Ref. 1.

For multilevel data transfers using random (direct)-access files, the CCC procedures have been implemented in ONEDANT. The standardized subroutines DOPC, CRED/CRIT, DRED/DRIT are used to effect multilevel data transfers using random-access files. A description of these procedures and subroutines is provided in Ref. 1.

5. Central Memory Restrictions. ONEDANT is designed to be operable within a 50 000 word central memory. At the same time it is easily adaptable to a larger amount of central memory for installations having a larger central memory.

6. Word Size. The code is designed to be easily converted from its basic long-word computer form to a form for use on short-word computers. (On a long-word computer a six character Hollerith word is a single-precision word while on a short-word computer it is a double-precision word.)

B. ONEDANT Code Package Structure

The ONEDANT code package consists of three major, functionally independent modules: an INPUT module, a SOLVER module, and an EDIT module. These modules are linked by means of binary interface files. The INPUT module processes any and all input specifications and data and, if required, generates the binary files for use by the SOLVER and/or EDIT modules. The SOLVER module performs the transport calculation and generates flux files for use by the EDIT module.

The SOLVER module also generates other interface files for use by other codes or for subsequent calculations by the SOLVER module. The EDIT module performs cross-section and response function edits using the flux files from the SOLVER module.

The interface files accepted, used, and generated by the modules are shown in Fig. 1. On the far left of the figure are indicated all the interface files accepted as input to ONEDANT while on the far right of the figure are shown all interface files that may exist at the end of a ONEDANT code execution.

A three-level overlay structure is used in ONEDANT for implementing the modules. Such a structure involves the use of a main overlay together with primary and secondary overlays.

The main (or 0,0) overlay contains the main program routine, which controls the calling of the primary overlays, together with those service subroutines used by more than one primary overlay.

The first overlay constitutes the INPUT module. It is structured into the first primary (or 1,0) overlay plus ten secondary overlays as shown in Fig. 2. Each of the secondary overlays performs a unique function, so that the INPUT module itself is constructed in a modular form.

The second overlay constitutes the SOLVER, or calculational, module. It consists of the second primary (or 2,0) overlay plus seven secondary overlays and is depicted in Fig. 3.

The third overlay is the EDIT module. It currently consists of the third primary, or (3,0), overlay plus a single secondary overlay.

A fourth overlay is used in ONEDANT. This overlay contains only the fourth primary, or (4,0), overlay with two subroutines. This fourth overlay provides highlights of the just-executed run as an aid to the user. These highlights are a printed summary of some of the pertinent facts, options, and decisions encountered during the run along with storage and run time information. This overlay is not considered to be a module in the sense of the first three overlays.

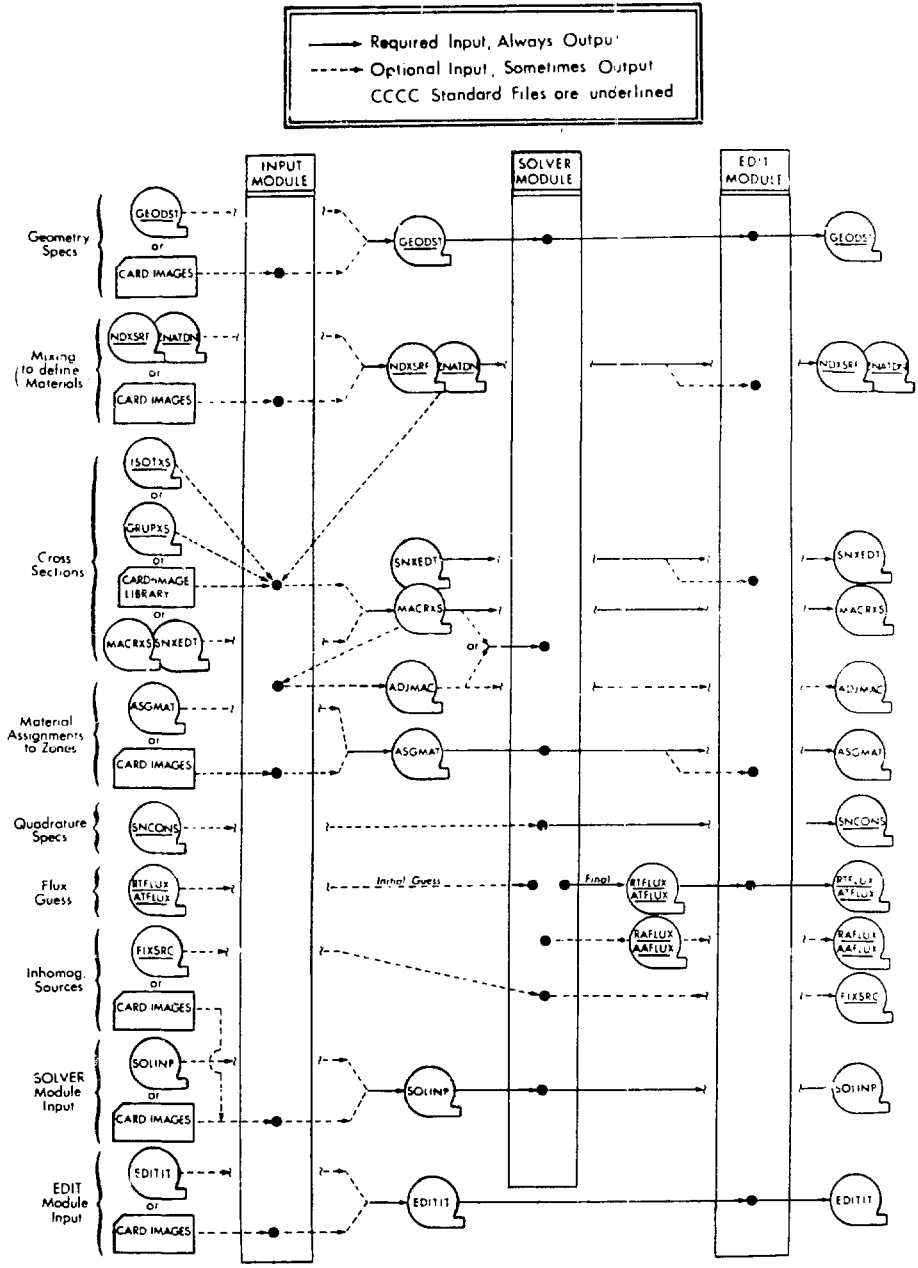


Fig. 1. Card-Image input and interface files accepted, used, and generated by ONEDANT.

C. INPUT Module

The INPUT module performs the necessary activities for processing all input data required for the execution of the SOLVER and/or EDIT modules. These activities include the reading of input data and the creation of binary interface files. The latter activity may require a certain degree of data processing. Each of these activities is discussed below.

In performing the reading-of-input-data activity, the INPUT module accepts standard interface files (binary), code-dependent binary interface files, or card-images for its input. These are shown in Fig. 1. As is indicated on the figure, input data to the code can be provided in several different forms and many combinations of forms to provide a great deal of flexibility to the user. Chapters IV and V provide specific information and further details on the specification of input data.

The second major activity in the INPUT module is the creation of binary interface files containing all input data. These files are subsequently used as the sole means of transmitting data to either the SOLVER or EDIT modules. The files emerging from the INPUT module are shown in Fig. 1 and take the form of either CCCC standard interface files or code-dependent interface files. In this file-creation activity the INPUT Module is called on to perform several types of tasks. As an example, the only form in which geometry-related information emerges from the INPUT module is in the form of a GEODST standard interface binary file. If a user supplies geometry-related input by means of card-image input, the INPUT module reads this input, translates the data into a GEODST-compatible form, and creates the resulting GEODST file. On the other hand, if the geometry-related information is supplied by the user through an already existing GEODST file, the INPUT module is required to do nothing. Fig. 1 indicates these two options and the resulting GEODST file. A second, more complex, example of the function of the INPUT module also serves to illustrate the use of Fig. 1. This example involves the mixing of isotopes, or nuclides, to create Materials which are subsequently assigned to physical regions in the problem (called Zones) to define the macroscopic cross-section data for the Zones. For this example it will be assumed that the user selects card-image input as the form for the INPUT module. First, the isotope mixing specifications appropriate for the desired Materials are input via card-image. The INPUT module reads this data, translates the data, and creates the two standard interface files NDXSRF and ZNATDN as shown in Fig. 1. These two files appear

as output from the INPUT module. Assuming next that the isotope cross sections are provided by the user as a card-image library, the INPUT module reads this library (in isotope-ordered form) and also reads the just-created NDXSRF and ZNATDN files. The mixing specifications provided by the latter files are applied to the isotopic cross-section data to generate Material cross sections, which are written, in group order, to a code-dependent binary file named MACRXS. (A group-ordered file named SNXEDT for use by the EDIT module is also created at this time but will not be considered in this example.) The MACRXS file becomes the sole source of cross-section data to the SOLVER module if the SOLVER calculation is to be a forward, or regular, calculation. If an adjoint calculation is to be performed by the SOLVER, the INPUT module re-reads the MACRXS file, performs the adjoint reversals on the cross sections, and creates the code-dependent binary file named ADJMAC containing the adjoint-reversed Material cross sections for use by SOLVER. These steps are all indicated schematically in Fig. 1.

D. SOLVER Module

The SOLVER module of ONEDANT has the function of effecting numerical solutions of the one-dimensional, multigroup form of the neutral-particle steady-state Boltzmann transport equation. The discrete-ordinates approximation is used for treating the angular variation of the particle distribution and the diamond-difference scheme is used for phase space discretization.⁴

In solving the transport equation numerically, an iterative procedure is used. This procedure involves two levels of iteration referred to as inner and outer iterations. The acceleration of these iterations is of crucial importance to transport codes in order to reduce the computation time involved. The ONEDANT SOLVER module employs the diffusion synthetic acceleration method developed by Alcouffe,⁵ an extremely effective method for accelerating the convergence of the iterations.

To display the iterative procedure and the application of the diffusion synthetic acceleration method, consider first the inner iteration equation for energy group g and inner iteration ℓ . Isotropic scatter is assumed only for simplicity. The basic inner iteration equation is written

$$\underline{\Omega} \cdot \nabla \tilde{\psi}_g^\ell(\underline{r}, \underline{\Omega}) + \sigma_g(\underline{r}) \tilde{\psi}_g^\ell(\underline{r}, \underline{\Omega}) = \sigma_{s, g \rightarrow g}(\underline{r}) \phi_g^{\ell-1}(\underline{r}) + Q Q_g(\underline{r}) \quad . \quad (1)$$

In Eq. (1), $\tilde{\psi}_g^\ell(r, \underline{\Omega})$ is the angular flux for group g at the ℓ th inner iteration using a scalar flux $\phi_g^{\ell-1}(r)$ assumed known at each inner iteration. QQ_g is the group source, which remains unchanged for the group throughout the performance of inner iterations. This group source contains scattering and fission contributions to the group together with any inhomogeneous source. The source is computed using the multigroup scalar fluxes and moments from the previous outer iteration. In the diffusion synthetic method, a corrected diffusion equation is used to determine the scalar flux ϕ_g needed for the next iteration. In actual fact, there are three separate schemes for writing the corrected diffusion equation to be used: the source correction scheme, the diffusion coefficient correction scheme, and the removal correction scheme. For the source correction scheme we write the corrected diffusion equation as

$$-\nabla \cdot D_g(r) \nabla \phi_g^\ell(r) + \sigma_{R,g}(r) \phi_g^\ell(r) = QQ_g(r) - R_g^\ell(r) \quad , \quad (2)$$

where

$$D_g(r) = 1/3\sigma_{tr,g}(r) \quad , \quad \sigma_{R,g}(r) = \sigma_g(r) - \sigma_{s,g \rightarrow g}(r)$$

and the correction term is

$$R_g^\ell(r) = \nabla \cdot \tilde{J}_g^\ell(r) + \nabla \cdot D_g(r) \nabla \tilde{\phi}_g^\ell(r) \quad . \quad (3)$$

In Eq. (3),

$$\tilde{\phi}_g^\ell(r) = \int d\Omega \tilde{\psi}_g^\ell(r, \underline{\Omega}) \quad , \quad \tilde{J}_g^\ell(r) = \int d\Omega \underline{\Omega} \tilde{\psi}_g^\ell(r, \underline{\Omega}) \quad . \quad (4)$$

Note that a tilde is used to indicate quantities calculated using the transport angular flux, $\tilde{\psi}_g^\ell$, while the scalar flux calculated from the corrected diffusion equation is without the tilde.

The source correction scheme for the inner iteration proceeds as follows: using $\phi_g^{\ell-1}$, known from the previous iteration, Eq. (1) is solved for $\tilde{\psi}_g^\ell$. This involves one sweep through the space-angle mesh. The correction term, R_g^ℓ , is then calculated using Eqs. (3) and (4) and, in turn, used in Eq. (2) to calculate ϕ_g^ℓ to complete one cycle or one inner iteration. The steps are repeated until suitable convergence is achieved. Note that for the first inner iteration for a group, a logical first guess for the scalar flux is obtained by solving Eq. (2) with $\ell = 0$ by setting R_g^0 to zero.

It is easy to show that if the iteration converges, it converges to the transport equation solution. Namely, drop all ℓ superscripts and set the transport scalar flux to the corrected diffusion scalar flux, $\tilde{\phi}_g = \phi_g$. Then substituting Eq. (3) into Eq. (4) yields

$$\nabla \cdot \tilde{J}_g(r) + \sigma_{R,g}(r) \tilde{\phi}_g(r) = QQ_g \quad ,$$

which is the converged transport balance equation obtained also by integrating Eq. (1) over all Ω .

The second level of iteration, the outer iteration, consists of one pass through the groups using Eqs. (1), (2), and (3) to obtain the group converged correction terms $R_g^k(r)$ and then to solve the multigroup corrected diffusion equation to generate new scalar fluxes consistent with a new fission source, if fission occurs in the problem. That is, the following multigroup diffusion equation is solved following the k th outer iteration:

$$\begin{aligned} -\nabla \cdot D_g(r) \nabla \phi_g^{k+1}(r) + \sigma_{R,g}(r) \phi_g^{k+1}(r) = Q_g(r) - R_g^k(r) \\ + \chi_g \sum_{g'=1}^G v \sigma_{f,g'}(r) \phi_{g'}^{k+1}(r) + \sum_{g' \neq g} c_{s,g' \rightarrow g}(r) \phi_{g'}^{k+1}(r) \quad . \quad (5) \end{aligned}$$

The source correction scheme outlined above for using the diffusion synthetic method is an effective scheme for inhomogeneous source problems. For eigenvalue problems, Eq. (5) must be homogeneous, and it is necessary to define a different scheme for the diffusion synthetic method. The diffusion coefficient correction scheme is one such scheme. In this scheme we redefine the corrected diffusion coefficient $\underline{D}_g(r)$ as

$$\underline{D}_g(r) = - \frac{\tilde{J}_g(r)}{\nabla \tilde{\phi}_g(r)} \quad , \quad (6)$$

so that $R_g(r) = 0$ for all r and g . Then, with $Q_g(r) = 0$ the inner iteration diffusion equation becomes

$$-\nabla \cdot \underline{D}_g^{\ell-1}(r) \cdot \nabla \phi_g^\ell(r) + \sigma_{R,g}(r) \phi_g^\ell(r) = QQ_g(r) \quad , \quad (7)$$

and the multigroup (outer iteration) diffusion equation becomes

$$\begin{aligned} -\nabla \cdot \underline{D}_g^k(r) \cdot \nabla \phi_g^{k+1}(r) + \sigma_{R,g}(r) \phi_g^{k+1}(r) &= \frac{\chi_g}{k_{\text{eff}}} \sum_{g'=1}^G v \sigma_{f,g'}(r) \phi_{g'}^{k+1}(r) \\ &+ \sum_{g' \neq g} \sigma_{s,g' \rightarrow g}(r) \phi_{g'}^{k+1}(r) \end{aligned} \quad (8)$$

where k_{eff} is the multiplication factor for the system. The same iteration procedure is used for this diffusion coefficient correction scheme as for the source correction scheme.

For eigenvalue problems the diffusion correction scheme has been found to accelerate the iterations as readily as the source correction scheme for inhomogeneous source problems. In fact, in ONEDANT, the diffusion coefficient correction scheme is used for inhomogeneous source problems in which fission and/or upscatter is present with the source correction scheme used only for inhomogeneous source problems with downscatter and no fission.

One disadvantage to the diffusion coefficient correction scheme is that infinite and negative diffusion coefficients are possible [see Eq. (6)]. If this occurs, Eq. (7) cannot be solved using current techniques. To overcome this difficulty, the removal correction scheme is employed. A corrected removal cross section is defined as

$$\tilde{\sigma}_{R,g}^k(r) \equiv \sigma_{R,g}(r) + R_g^k(r)/\phi_g^k(r) \quad (9)$$

where $R_g^k(r)$ is defined by Eq. (3). With this, the diffusion synthetic method is modified and Eq. (2) becomes

$$-\nabla \cdot D_g(r) \nabla \phi_g^{\ell}(r) + \tilde{\sigma}_{R,g}^{\ell-1}(r) \phi_g^{\ell}(r) = Q_g^{\ell}(r)$$

and Eq. (5) becomes

$$-\nabla \cdot D_g(r) \nabla \phi_g^{k+1}(r) + \tilde{\sigma}_{R,g}^{k+1}(r) \phi_g^{k+1}(r) = \frac{\chi_g}{k_{\text{eff}}} \sum_{g'=1}^G \nu \sigma_{f,g'}(r) \phi_{g'}^{k+1}(r) + \sum_{g' \neq g} \sigma_{s,g' \rightarrow g}(r) \phi_{g'}^{k+1}(r) \quad (10)$$

The iteration procedure is entirely analogous to that for the diffusion coefficient correction scheme and, again, if it converges, it converges to the transport balance equation solution. This removal correction scheme is employed in eigenvalue problems or source problems with fission and/or upscatter only when the diffusion coefficient correction scheme produces negative or infinite diffusion coefficients.

As outlined above, the function of the SOLVER module is to effect numerical solutions to the one-dimensional steady-state, multigroup form of transport equation using the diffusion synthetic method to accelerate the iterative convergence rate. The module is essentially a free-standing entity, and input to and output from the module is in the form of binary files together with limited printed output. The binary interface files used as input to the SOLVER module are shown in Fig. 1. The files required for execution of the module are a GEODST standard interface file together with the code-dependent interface files MACRXS or ADJMAC, ASGMAT, and SOLINP. Optional files, which may be input to the SOLVER module, are the standard interface files SNCONS, RTFLUX or ATFLUX, and FIXSRC.

The output from the SOLVER module always consists of the scalar flux standard interface file RTFLUX (or ATFLUX if an adjoint problem were run), the standard interface file SNCONS, and user-selected printed output. If desired by the user, the angular flux standard interface file RAFLUX (or AAFLUX, if an adjoint problem were run) will be produced. If an inhomogeneous source problem were run, a FIXSRC standard interface file would be produced.

E. EDIT Module

The function of the EDIT Module is to produce the printed edit-output selected by the user. Edit-output refers to information which is obtained from data contained on one or more interface files but which generally requires manipulating or processing of the data. An example of the edit-output is a microscopic reaction-rate distribution, $\sigma\phi$, where σ is a particular multigroup, microscopic cross section for a particular isotope or nuclide and ϕ is the multigroup scalar flux distribution obtained from the SOLVER module. In this example, data from both a cross-section interface file and a scalar flux file are required to be recovered, multiplied, and the product printed.

INPUT Module Driver OVERLAY (1,0)									
(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,10)
Setup & Storage	Geometry	Card-Image Library Pre-Processor	Mixing Specs	GRUPXS Library Processor	ISOTXS Library Processor	Card-Image Library Processor	SOLVER Input	EDIT Input	Adjoint Reversal

Fig. 2. Overlay structure of the INPUT module.

SOLVER Module Driver OVERLAY (2,0)						
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)
Initialize	S_n Constants	Flux Guess, Q-Source	Computational Data Preparation	Outer Iteration	Final Printing	Binary File Output

Fig. 3. Overlay structure of the SOLVER module.

The EDIT module is an essentially free-standing module accepting only interface files as input and producing printed output. The required input files for execution of the EDIT module are the code-dependent binary interface file EDITIT and the standard interface files RTFLUX (or ATFLUX) and GEODST as shown in Fig. 1. Optional input files are the standard interface files NDXSRF and ZNATDN and the code-dependent files SNXEDT and ASGMAT. The code-dependent files are produced by the INPUT module.

III. CARD-IMAGE INPUT FORMAT RULES

This chapter describes the various rules, restrictions, and options available to the user when creating the input for ONEDANT. First are described the details associated with free-field input since most users will likely select this form. Next is presented the information needed for user-specified input formats followed by information for fixed-field FIDO input.

A. Free-Field Input

1. Card-Image Ground Rules

- (a) Eighty (80) columns available.
- (b) No special columns, i.e., no column is treated any differently than any other column.

2. Delimiters (Separators) and Terminators

- (a) Data Item Delimiter (Separator): one or more blanks, a comma, or end of card.

Note: Hereafter when an item is referred to as being delimited, e.g., delimited T, it means that the item must be separated from other data items by a blank, comma, or end of card.

- (b) Card Terminator: Slash (/), delimiting not required. All entries on a card beyond the slash are ignored.
- (c) Block Terminator: Delimited T. Information beyond the T on the card will be ignored.
- (d) Array Terminator: New array name or Block Terminator.
- (e) String Delimiter: Semicolon (;), delimiting not required on; .
- (f) String Terminator: Semicolon or new array name or Block Terminator.
- (g) Data Item Terminator: Data item delimiter (Separator) or any of the above Terminators.

3. Numerical Data Item Ground Rules

- (a) Must not contain embedded blanks.
- (b) Must not contain any nonnumeric characters except for E (for exponent), decimal point, or plus and minus signs.

4. Hollerith Data Item Ground Rules

- (a) Must begin with alphabetic character [see (c) below for exception] and may contain from one to six characters.
- (b) Must not contain any of the following characters: =, \$, *, blank, comma, slash, semicolon, double quote ("). [see (c) below for exception.]
- (c) Hollerith data records may be delimited with double quotes to override (a) and (b) restrictions. For example, PU/239 is not allowed but "PU/239" is allowed.

5. Array Identification and Ordering

- (a) To identify an array for which data entries are to be made, one simply enters the appropriate array name, followed by an equal (=) sign (no space between name and =) and then enters the desired data, e.g.,
CHI= 0.95, 0.10 0.05 0.0 .
- (b) Within a given Block, arrays may be entered in any order.

6. Block Identification and Ordering

- (a) No explicit Block identification is required. Array identification is sufficient to tell the code which Block is involved. Recall, however, that a Block Terminator (delimited T) must be entered when all input arrays for a given Block have been entered.
- (b) Blocks must be ordered.

7 Input Data Operators. Several data operators are available to simplify the input. Most of these operators are FIDO operators but several are new and represent extensions to FIDO.

IMPORTANT NOTE: The following data operators can only be used with arrays containing integer, real, or a combination of integer/real data entries. They are NOT usable with arrays that may contain Hollerith data items.

In free-field the data operators are specified in the general form

n Ø d

where

n is the "data numerator", an integer or blank,
Ø is any one of several "data operators" described in Table I, and
d is a "data entry" (may be blank for some operators).

NOTE: When a "data numerator" is required with a "data operator," there must be no space between the data numerator and the data operator. There may be any number of blanks between the data operator and the "data entry" if the latter is required.

In entering data using data operators, it is convenient to think of an index or pointer that is under the control of the user and which specifies the position in the data string into which the next data item is to go. The pointer is always positioned at string location number 1 when either an array identifier or a string terminator (;) is entered.

B. User-Specified Input Formats

If desired, input data for an array can be provided in a format specified by the user. To specify the format for the input data to an array, the user can use the characters U or V as follows:

- U Operator - (1) Enter the array identification and follow this with a delimited U.
(2) On the next card-image enter the desired format enclosed in parentheses anywhere in columns 1-72.
(3) On the next and succeeding cards enter the data using ordinary FORTRAN rules.

Example: CHI= U
(6E12.5)
data in 6E12.5 format

V Operator - has same effect as U except the desired format is not entered, instead the format read in the last preceding U array is used.

TABLE I

FREE-FIELD DATA OPERATORS

In the table below an entry of ~ for either the data numerator or data entry indicates that the item is not required for the particular data operator.

<u>Data Numerator</u>	<u>Data Operator</u>	<u>Date Entry</u>	<u>Remarks</u>
n	R	d	REPEAT OPTION: Enter the data entry d n successive times in the current data string. Example: 3R 0.0 → 0.0 0.0 0.0
n	I	d	LINEAR INTERPOLATE OPTION: Enter the value d into the data string followed by a n interpolated entries equally spaced between d and the next <u>data entry</u> . Allowed for both real and integer data although the spacing between interpolated integer data points must be integer. Example: 3I1, 5 → 1 2 3 4 5 but 3I1, 4 will cause an error if the array data type is integer.
n	L	d	LOGARITHMIC INTERPOLATE OPTION: The effect is the same as that of "I" except that the resulting interpolates are equally separated in log-space.
~	F	d	FILL OPTION: Fill the remainder of the data string with the value d.
n	Z	--	ZERO OPTION: Enter the value zero in the data string n successive times. Example 4Z → 0 0 0 0
n	S	~	SKIP OPTION: Causes the pointer to skip n positions in the current string leaving the data values in those positions unchanged.
n	A	~	POSITIONING OPTION: Set the pointer in the current data string to the <u>nth</u> data item position in that string. Example: Suppose the array SOURCX is to be a single string of length 100, all values of which are 0.0 except the 15 th value which is to be unity. This can be entered simply as SOURCX= 100Z 15A 1.0

TABLE I (cont.)

<u>Data Numerator</u>	<u>Data Operator</u>	<u>Date Entry</u>	<u>Remarks</u>
n	Q	m	SEQUENCE REPEAT OPTION 1: Enter the last m data entries into the current string in sequence n successive times. Example: 1 2 3 1 2 3 1 2 3 can be input at 1 2 3 2Q3
n	G	m	SEQUENCE REPEAT OPTION 2: Same effect as "Q" except the sign of each entry in the sequence is reversed each time the sequence is repeated. Example: 1 2 -1 -2 1 2 can be input as 1 2 2G2.
n	N	m	SEQUENCE REPEAT OPTION 3: Same effect as "Q" except the order of the sequence is reversed each time the sequence is entered.
n	M	m	SEQUENCE REPEAT OPTION 4: Same effect as "N" except that the sign of each entry in the sequence is reversed each time the sequence is entered. For example: 1 2 3 -3 -2 -1 1 2 3 can be input as 1 2 3 2M3.
n	Y	m	STRING REPEAT OPTION: Enter the preceding <u>m strings</u> of data into the current array n successive times. (For multistring arrays only).
n	X	-	COUNT CHECK OPTION: Causes code to check the number of data items entered into the current string to see if the number of items equals n. If count is not correct, an error message will be printed, an attempt will be made to continue processing all remaining input, and then the problem will be halted. (Error diagnostic aid.)
-	E	-	END OF STRING OPTION: Causes pointer to skip to the end of the current data string leaving values of skipped data items unchanged.

C. Fixed-Field FIDO Input

1. Card-Image Ground Rules

- (a) Seventy two (72) columns available.
- (b) Each card divided into six "fields" of 12 columns each.
- (c) Each 12-column "field" subdivided into three subfields containing 2, 1, and 9 columns, respectively. Hereafter these subfields will be referred to as Subfield 1 (the first two columns of each field), Subfield 2 (the third column in each field), and the Data Subfield (the remaining nine columns in each field).

2. Delimiters (Separators) and Terminators

- (a) Data Item Delimiter (Separator): Field and subfield column boundaries.
- (b) Card Terminator: Slash (/) in second subfield with first subfield blank. All entries following the slash on the card are ignored.
- (c) Block Terminator: T in second subfield of any field. All entries beyond the T on that card are ignored.
- (d) Array Terminator: New array identified in first and second sub-fields of next field, or a Block Terminator.
- (e) String Delimiter: Semicolon (;) in second subfield of any field. Data subfield of that field is ignored.
- (f) String Terminator: Semicolon or Array Terminator or Block Terminator.

3. Numerical Data Ground Rules

- a. Standard FORTRAN convention.
- b. Data items entered in third subfield (the data subfield) in each field only.

4. Hollerith Data Item Ground Rules

Hollerith data not allowed.

5. Array Identification and Ordering

- (a) To identify an array for which data are to be entered in the fixed-field FIDO format, one simply enters the array number (integer, ≤ 99) in the first subfield of any field followed by the array-type indicator (array purpose character) in the second subfield. If the array data is integer (fixed point), the array-type indicator is the dollar sign (\$); if the array data is real (floating point) the array-type indicator is an asterisk (*). The third subfield is left blank.
- (b) Arrays may be entered in any order within a given Block.

6. Block Identification and Ordering

- (a) No explicit Block identification required. Array identification is sufficient to tell the code which Block is involved. Recall that a Block Terminator (T) must be entered when all input arrays for a given Block have been entered.
- (b) Blocks must be ordered.

7. Input Data Operators. The fixed-field FIDO data operators are the same operators used in the free-field input shown in Table I. In fixed-field FIDO usage of these operators, however, the following rules must be observed:

- (i) the "data numerator", if required, must be entered in the first subfield of a field,
- (ii) the "data operator" must be entered in the second subfield of a field, and
- (iii) the "data entry", if required, must be entered in the third subfield of a field.

IV. ONEDANT CODE PACKAGE INPUT SPECIFICATIONS

Card-image input to the ONEDANT code package consists of (i) Title Card Control Input (always required) and subsequent Title Cards, and (ii) up to six BLOCKS of input data with each BLOCK pertaining to a specific class of input.

This chapter contains a brief overview of the specification of input followed by a Mini-Specification Sheet to be used as a quick reference for the input BLOCKS and their contained data arrays. The third section provides a relatively detailed description of the input specifications.

A. Overview of the Specification of Input

The ONEDANT card-image input consists of a Title Card section followed by six blocks of input. The blocks are:

<u>BLOCK</u>	<u>NAME</u>	<u>PURPOSE</u>
I.	Controls & Dimensions	Provides basic parameters associated with the physical problem model for purposes of data storage requirements. Also provides special code execution controls.
II.	Geometry	Provides a description of the physical problem model geometry.
III.	Cross Sections	Provides the input Isotopic microscopic cross sections.
IV.	Mixing	Provide mixing instructions for forming macroscopic Material cross sections from the input Isotopic microscopic cross sections, and provides instructions for assigning the macroscopic Materials to the Zones of the physical problem model.
V.	Solver Input	Provides input specifications and data for the particular calculation to be effected by the one-dimensional, multigroup, discrete-ordinates, diffusion synthetic accelerated Solver module.
VI.	Edits	Provides information that allows editing of the fluxes output by the SOLVER module and that causes subsequent edit calculations using those fluxes, together with cross sections, response functions, etc., to be performed.

BLOCKS MUST APPEAR IN THE INPUT STREAM IN THE ABOVE ORDER

The Title Card section and BLOCK I input are always required in every run; the remaining blocks are required only as necessary for the particular run desired. For example, the input associated with BLOCK II (Geometry) may be omitted if the geometry specifications are being supplied from a previously created GEODST standard interface file.

BLOCKS II, III, and IV are all directly associated with the INPUT module of ONEDANT. BLOCK V is associated solely with the SOLVER module, which effects the solution of the multigroup, discrete-ordinates, one-dimensional, steady-state, diffusion synthetic accelerated transport equation. BLOCK VI is associated uniquely with the EDIT module of ONEDANT.

BLOCKS consist of one or more ARRAYS and a BLOCK TERMINATOR, the character "T." Each block is identified solely by the arrays comprising the block so that the presence of one or more of these arrays in the input establishes the existence of that block of input. Arrays may be entered in any order within a given block. Blocks, however, must appear in ascending numerical order. If a block is to be omitted, everything associated with that block, including the terminal T, is omitted.

ARRAYS are comprised of one or more DATA STRINGS, each of which contains one or more DATA ITEMS. The majority of the ONEDANT input arrays contain only a single string and can be thought of as one-dimensional arrays or vectors. The term STRINGED ARRAY refers to any array containing more than one string of data, i.e., a multidimensional array. Arrays are identified by either a unique array Hollerith name (up to six characters) or an array number. Input to a given array is terminated by either (i) the appearance of a new array name or number or (ii) the appearance of the block terminator, T, in the input stream.

In STRINGED ARRAYS, data strings are delimited by the special STRING DELIMITER, the semicolon (;).

Data items are separated (delimited) by either (i) a blank, (ii) a comma, or (iii) the end of a "card." A data item is terminated by any of the separators above or by the semicolon, an array name (or number), a block terminator, T, or the slash, /.

Card-image input to ONEDANT is processed by a very flexible input routine that supports a variety of input forms. One of the forms is the

FIDO input form used in numerous existing codes. Users who are familiar with this form may thus readily produce input for ONEDANT without having to learn a new input form. Available also is a free-field capability particularly useful for input via remote terminals. Associated with the free-field feature is the extension of FIDO-like options to include both hollerith and mixed data-type input. With this capability it is possible to supply input containing real, integer, and Hollerith data in the same string.

A brief description of the manner in which input can be supplied follows. A full description of the options available is presented in Ch V.

For each array there is associated both a Hollerith name (up to six alphanumeric characters) and a number. Either name or number uniquely identifies the array for which input is to be supplied. To distinguish the array name or number from subsequent data items there is appended to the name or number an array identifier. For free-field input the array identifier suggested is the equal sign (=). This character can be used for any data-type input, real, integer, Hollerith, or any combination, so long as the data is entered in free-field format. If the array's data is entirely integer data and is to be entered in fixed-field FIDO format, the array identifier for that array is the dollar (\$) sign; for a real data array input in fixed-field FIDO format, the asterisk (*) array identifier character is to be used. As an example, consider the input for specifying the coarse-spatial-mesh boundaries. In ONEDANT the array name for this input is XMESH, a BLOCK II array. If the desired mesh boundaries are to be at 0.0, 10.0, 20.0, and 30.0 cm, the data could be entered as:

```
XMESH= 0, 10, 20, 30 .
```

Options are available to repeat data items, to produce numbers interpolated between two numeric data items, and to perform other more sophisticated operations designed to reduce the volume of input. The interpolate

option, for example, can be used in the preceding XMesh example input where two equally spaced interpolation points can be placed between the entries 0.0 and 30.0. Thus, the above example can be input as:

```
XMesh= 2I 0, 30 ,
```

where the data operator I denotes interpolation. As another example of a commonly used operation, the repeat option, suppose that in the array named LBEDO one wishes to enter the value 0.7 twelve successive times. This can be accomplished by entering

```
LBEDO= 12R 0.7 ,
```

where the data operator R denotes repeated entry.

The full list of available operations is described in Chapter III.

One additional option for the user available in ONEDANT involves the inputting of data to certain arrays whose data members are a collection of single-word, independent control parameters. For these arrays the user can either input the collective array name followed by the data entries for the control parameters comprising that particular collective array, or the user can input the individual array members by their unique member names individually. As an example, the collective array named SOLIN in BLOCK V contains the 5 control parameters IEVT, ISCT, ITH, IBL, and IBR. Suppose that IEVT = 2, ISCT = 3, ITH = 1, IBL = 1, IBR = 0. Input can be provided as either

```
SOLIN= 2 3, 1, 1, 0 (using collective array name, SOLIN)
```

or

```
IEVT= 2, IBR= 0 , IBL= 1 , ISCT= 3 , ITH= 1.
```

In the input specifications that follow, when the contents of an array have listed named members, input may be effected using either the collective array name or the individual member names.

B. ONEDANT INPUT: Mini-Specification Sheet

Comments:

- Collective arrays, e.g., DIMENS in BLOCK I, may be specified in the input by either using the collective array name, followed by =, followed by the requisite number of data entries, or by using the names of the individual data members, each followed by = and the data value, e.g., IGEOM= 1, NGROUP= 42.
- Entries within brackets, [], indicate the number of entries in the array. If the entry is of the form [I;J], then the array consists of J strings of data each of which contains I data words. [-] means that the number of entries is arbitrary.
- For those arrays amenable to fixed-field FIDO input, the appropriate array number and FIDO array-type indicator is shown in parenthesis below the array name.

1 **TITLE CARD CONTROL**
(316 Format)
NHEAD,NOTTY,NOLIST

2 **TITLE CARDS**
(if NHEAD > 0)

3 **BLOCK I: CONTROLS & DIMENSIONS**

DIMENS [8] or IGEOM
= NGROUP
ISN
NISO
MT
NZONE
IH
IT

MAXLCH [1]

NOEXEC [3] or NOFGEN
= NOSOLV
NOEDIT

NOFILE [7] or NOGEOD
= NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM T

4 **BLOCK II: GEOMETRY**

KMESH [IM + 1]
(1*)

XINTS [IM]
(4\$)

ZONES [IM]
(7\$) T

5 **BLOCK III: CROSS SECTIONS**

LIB [1]

LNK [1]

Following arrays used only if
LIB=ODNINP, XSLIB, OR BXSLIB

CARDS [9] or MAXORD
= IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD

NAMES [NISO]

EDNAME [IHT-3]

NTP1 [NISO]
(10 \$)

VEL [NGROUP]
(11*)

EBOUND [NGROUP + 1]
(12*) T

5a **Iff LIB= ODNINP**
insert BCD card-image
cross sections here.

6 **BLOCK IV: MIXING**

PREMIX [-;-]

MATLS [-;MT]

ASSIGN [-;NZONE]

ASGMOD [-;-]

CMOD [1]

MATNAM [MT]

ZONNAM [NZONE] T

C. Input Specifications

CARD 1: TITLE CARD CONTROL

{Always Required}

Format 3I6

<u>Word</u>	<u>Variable Name</u>	<u>Comments</u>
1	NHEAD	Number of Title (header) cards to follow
2	NOTTY	Selected output to on-line user terminal? 0/1 = Yes/No (Default=0)
3	NOLIST	Listing of all card-image input with the output? 0/1 = Yes/No (Default=0)

CARDS 2 through NHEAD+1: TITLE CARDS

{Required if NHEAD > 0}

Format 12A6

NHEAD Title Cards containing descriptive comments about the ONEDANT run.

BLOCK I: CONTROLS & DIMENSIONS

{Always Required}

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS		
Name	Number	[Length]	Word	Name	Comments
		{Present if}			
DIMENS	-	[8]		Basic parameters for determining storage
		{Always}		1. IGEOM	Geometry: 1/2/3 = Plane/Cylind./Sphere
				2. NGROUP	Number of energy groups
				3. ISN	S _N Angular quadrature order (even number)
				4. NISO	No. of "isotopes" on basic input cross-section library
				5. MT	No. of materials to be created
				6. NZONE	No. of geometric zones in problem (each neutronically homogeneous)
				7. IM	No. of coarse spatial mesh intervals
				8. IT	Total no. of fine spatial mesh intervals
				NOTE:	A positive integer must be entered for each of the above parameters.
MAXLCM	-	[1]		Length of Large Core Memory (LCM) desired (Default = 50000 ₁₀)
		{Optional}			
<u>Note:</u> THE FOLLOWING ARRAYS PROVIDE SPECIALIZED OPTIONS FOR THE MORE ADVANCED USER (See Ch. VIII).					
NOEXEC	-	[3]		Execution suppression flags.
		{Optional}		1. NOFGEN	Suppress any further execution of the INPUT Module, i.e., generate <u>no</u> files (all desired files exist): 0/1 = No/Yes (Default = 0)
				2. NOSOLV	Suppress execution of SOLVER Module: 0/1=No/Yes (Default = 0)
				3. NOEDIT	Suppress execution of EDIT Module: 0/1=No/Yes (Default = 0)
NOFILE	-	[7]		Interface file suppression flags.
		{Optional}		1. NOGEOD	Suppress the generation of a GEODST file even though BLOCK II card input is present: 0/1 = No/Yes (Default = 0)
				2. NOMIX	Suppress the generation of the NDXSRF and ZNATDN standard interface files even though mixing specification input exists: 0/1 = No/Yes (Default = 0)
				3. NOASG	Suppress the generation of the ASGMAT code-dependent interface file even though BLOCK IV material-to-zone assignment input exists: 0/1 = No/Yes (Default = 0)

BLOCK I (continued)

- 4. **NOMACR** Suppress the generation of the MACRXS and SNXEDI code-dependent interface files even though cross-section and mixing specification input exists: 0/1 = No/Yes (Default = 0)
- 5. **NOSLNP** Suppress the generation of the SOLINP code-dependent interface file even though BLOCK V input exists: 0/1 = No/Yes (Default = 0)
- 6. **NOEDTT** Suppress the generation of the EDITIT code-dependent interface file even though BLOCK VI input exists: 0/1 = No/Yes (Default = 0)
- 7. **NOADJM** Suppress the generation of the ADJMAC code-dependent interface file even though an adjoint calculation is called for: 0/1 = No/Yes (Default = 0)

BLOCK II: GEOMETRY

{Required unless on existing GEODST file is to be used}

<u>ARRAY SPECIFICATION</u>		<u>ARRAY CONTENTS AND DESCRIPTIVE COMMENTS</u>	
<u>Name</u>	<u>Number [Length]</u>	<u>Word</u>	<u>Name</u> <u>Comments</u>
	{Present if}		
XMESH	, 1* [IM+1] {Always}	Coarse spatial mesh interval boundaries
XINTS	, 4\$, [IM] {Always}	Number of equally spaced fine-mesh intervals in each coarse-mesh interval
NOTE: $\sum_{I=1}^{IM} XINTS(I) = IT$			
ZONES	, 7\$, [IM] {Always}	Zone number for each coarse-mesh interval (The number 0 (zero) may be used to specify that a coarse-mesh interval is a pure void. A "0" does not count as a zone.)

BLOCK III: CROSS SECTIONS

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS		
Name	Number	[Length]	Word	Name	Comments
LIB	-	[1]		Source of cross-section data. Enter one of the following Hollerith names:
{Always}				GRUPXS	(Group-ordered standard interface file)
				ISOTXS	(Isotope-ordered standard interface file)
				ODNINP	(Card-image BCD library supplied immediately following this BLOCK III)
				XSLIB	(Card-image BCD library supplied as a separate file named XSLIB)
				BXSLIB	(Binary form of card-image, BCD library as a separate file named BXSLIB)
				MACRXS	(Use existing files named MACRXS for SOLVER Module, SNXEDT for EDIT Module. Under this option any remaining BLOCK III input and, unless otherwise specified in BLOCK I, any PREMIX and MATLS input in BLOCK IV will be ignored)
LNG	-	[1]		Number of the last neutron group in a coupled neutron-photon library.
{Optional}					

NOTE: THE REMAINING ARRAYS IN BLOCK III ARE USED ONLY IF THE SOURCE OF CROSS-SECTION DATA IS ODNINP OR XSLIB.

BLOCK III, continued

ARRAYS BELOW USED ONLY IF LIB= ODNINP OR LIB= XSLIB

ARRAY SPECIFICATION		ARRAY CONTENTS AND DESCRIPTIVE COMMENTS	
Name	Number [Length]	Word	Name Comments
	{Present if}		
CARDS	- [9] {Required}	Basic parameters for card-image BCD cross-section libraries. (Refer to Chapter V for details)
		1. MAXORD	Highest Legendre order in scattering tables {Required}
		2. IHM	Number of rows in a cross-section table {Required}
		3. IHT	Row number of total cross section in table {Required}
		4. IHS	Row number of self-scatter cross section in table {Default = IHT + 1}
		5. IFIDO	Format of cross-section library: 0/1/2 = Los Alamos (DTF)/Fixed-Field FIDO/Free-Field FIDO {Default = 0}
		6. ITITL	Title card precedes each table? 0/1 = No/Yes {Default = 0}
		7. I2LP1	Higher order scatter cross sections contain the 2L+1 factor, where L = Legendre scattering order? 0/1 = No/Yes {Default = 0}
		8. SAVBXS	Create and save binary form of card-image library as file BXSLIB? 0/1 = No/Yes {Default = No}
		9. KWIKRD	Process FIDO-format, card-image, BCD library with fast processor at the sacrifice of error checking? 0/1 = No/Yes {Default = Yes}
NAMES	- [NISO] {Optional}	One to six character, left-justified Hollerith name for each isotope in the library {Default = ISOn where n is the integer position of the isotope in the library}
EDNAME	- [IHT-3] {Optional}	One to six character, left-justified Hollerith name for the edit positions in the cross-section table (those positions preceding σ_a) {Default = EDIT1, EDIT2, etc.}
NTPI , 10\$,	[NISO] {Required if scattering order varies with Isotope}	Number of Legendre scattering orders for each isotope in the library. {Default = MAXORD + 1}
VEL , 11* ,	[NGROUP] {Required if doing α calculation; see IEVT}	Particle speed for each energy group
EBOUND , 12* ,	[NGROUP+1] {Optional}	Energy boundaries for each group

BLOCK IV: MIXING
{Normally Required}

Input in this block describes the nuclide mixing that ultimately creates the macroscopic cross-section sets that are assigned to the Zones in the physical problem model.

The general procedure involves first mixing the Isotope cross-section data from a basic library (defined in BLOCK III) to create Material cross-section sets and then to assign one or more Materials to each Zone to define the macroscopic data for that Zone.

Material cross sections are "permanently stored" in energy-group order and are the only cross-section data available to either the SOLVER or EDIT Modules. These Materials are defined by the MATLS array described below.

For the user's convenience, optional temporary mixtures (or premixes) can be created. These temporary mixtures are defined by the PREMIX array described below and can be used as components of Materials. The temporary mixtures are indeed temporary; they are not stored and are thus forgotten by the code once the Materials have been created.

The assignment of Materials to Zones is achieved through the ASSIGN and ASGMOD arrays described below. Additional optional input through the arrays MATNAM, ZONNAM, and CMOD is also described.

PREMIX (Temporary Mixture) SPECIFICATIONS
{Optional}

FORM: PREMIX= PREID_a COMP₁ DEN₁ COMP₂ DEN₂ ... ;
PREID_b COMP₃ DEN₃ COMP₄ DEN₄ ... ;

where

PREID_a, PREID_b, etc., are unique Premix identifiers. IT IS STRONGLY RECOMMENDED THAT PREMIX IDENTIFIERS BE ONE TO SIX CHARACTER HOLLERITH NAMES BEGINNING WITH AN ALPHABETIC CHARACTER!

COMP₁, COMP₂, etc., are unique identifiers for the constituents or components of the premix being specified. A component identifier may refer to either an isotope or to another premix (a premix may not be a component of itself). If the component identifier refers to an isotope from the basic input cross-section library, the identifier may either be (i) the Hollerith name of the isotope or (ii) an integer, $I(1 \leq I \leq \text{NISO})$, in which case I refers to the I th isotope on the basic cross-section library. If the component identifier refers to a premix, the identifier should be the same as the premix identifier used when specifying that premix.

DEN₁, DEN₂, etc. are the atom densities, volume fractions, etc., associated with the immediately preceding components. NOTE THAT SEMICOLONS (;) MUST SEPARATE THE SPECIFICATIONS FOR EACH PREMIX!

BLOCK IV, continued

MATERIAL SPECIFICATIONS

{Normally* Required}

Material specifications are input via the MATLS array in either of the two following forms:

FORM 1: MATLS= ISOS

With this abbreviated form, "Isotopes" from the basic input cross-section library (see BLOCK III) are directly designated as Materials such that the first Isotope on the library becomes the first Material, the second Isotope becomes the second Material, etc. Similarly, the Hollerith Isotope names are directly used for the Hollerith Material names.

NOTE: With this form, the number of Materials, MT, will normally be equal to the number of isotopes, NISO. If $MT \neq NISO$, then $MT < NISO$ is REQUIRED!

FORM 2: MATLS= MATID_a COMP₁ DEN₁ COMP₂ DEN₂ ... ; MATID_b COMP₃ DEN₃ COMP₄ DEN₄ ... ; etc.

where

MATID_a, MATID_b, etc., are unique identifiers for each of the MT Materials to be specified. The MATID's may be input as either (i) unique Hollerith names (one to six characters beginning with an alphabetic character), or (ii) a unique integer in the range 1 to MT, inclusive. If Hollerith names are used for the Material identifiers, the first named Material is indexed as Material 1, the second named Material is indexed as Material 2, etc. If integers are used for the Material identifiers, the integer denotes the index or position of the Material in the ordered list of MT (total number of) Materials. Additionally, if integers are used for the

*see Chapter V.D. for exceptions

BLOCK IV, continued

MATID's, Hollerith names for the Materials may be optionally provided by the user via the MATNAM array described below. (If names are not provided, the default Hollerith Material name MATn will be created by the code where n is the integer Material identifier.)

NOTE: All Material identifiers (MATID's) must be of like form (Hollerith name or integer).

COMP₁, COMP₂, etc., are unique identifiers identifying either Isotopes from the basic library or Premixes to be used as components or constituents of the Material identified by the immediately preceding MATID. If the desired component is an Isotope, then the component identifier, COMP, may either be (i) the unique Hollerith name associated with the Isotope or (ii) a unique integer in the range 1 to NISO denoting the index or position of the desired Isotope in the ordered list of Isotopes in the basic library. If the desired component is a Premix, then the component identifier, COMP, should exactly match the Premix identifier used in the PREMIX array specifications.

DEN₁, DEN₂, etc., are the atom densities, volume fractions, etc. associated with the immediately preceding components.

NOTE 1: EACH MATERIAL SPECIFICATION MUST BE SEPARATED FROM THE NEXT BY A SEMICOLON (;).

NOTE 2: THERE MUST BE EXACTLY MT MATERIALS SPECIFIED.

FORM 2 SPECIAL FEATURE: The Material specification string characterized above by MATID COMP₁, DEN₁, COMP₂ DEN₂ ... ; can be input in the abbreviated form

MATID COMP ;

if the Material identified by MATID contains only a single component (Isotope or Premix) identified by COMP, the component identifier, with a density of 1.0.

ASSIGNMENT OF MATERIALS TO ZONES

{Normally* Required}

The assignments of one or more Materials to each Zone in the physical problem model are made via the ASSIGN array. For concentration searches and some other possible uses, additional assignment information can be input via the ASGMOD array.

THE ASSIGN ARRAY {Required if SOLVER or EDIT Modules to be executed}

There are two basic forms for using the ASSIGN array.

FORM 1: ASSIGN= MATLS

With this abbreviated form, Materials as defined in the MATLS array are directly assigned to Zones such that the first Material is assigned to the first Zone, the second Material to the second Zone, etc. Similarly, the Hollerith Material names are used directly for the Hollerith Zone names. NOTE: With this form, the number of Zones, NZONE, will normally be equal to the number of Materials, MT. If NZONE \neq MT, then NZONE < MT is REQUIRED!

FORM 2: ASSIGN= ZONID_a MATID₁ CONC₁ MATID₂ CONC₂ ... ; ZONID_b
MATID₃ CONC₃ MATID₄ CONC₄ ... ; etc.

Here

ZONID_a, ZONID_b, etc., are unique Zone identifiers for each of the NZONE Zones. The ZONID's may be input either as (i) unique Hollerith names (one to six characters beginning with an alphabetic character), or (ii) a unique integer in the range 1 to NZONE, inclusive. If Hollerith names are used for the Zone identifiers, the first named Zone is given a Zone number of 1, the second named Zone is given a Zone number of 2, etc. If integers

*See Chapter V.E. for exceptions.

are used for the ZONID's, the integer denotes the Zone number, and its value must lie in the range 1 to NZONE, inclusive. Additionally, if integers are used for the ZONID's, Hollerith names for the Zones may be optionally provided by the user via the ZONNAM array described below. (If names are not provided, the default Hollerith name ZONEn will be used, where n is the integer Zone identifier, i.e., Zone number.)

NOTE 1: All Zone identifiers (ZONID's) must be of like form (Hollerith name or integer).

NOTE 2: The Zone numbers associated with the ZONID's are the same numbers used in the ZONES array of BLOCK II.

MATID₁, MATID₂, etc., are the desired Material identifiers for the Materials to be used as components or constituents of the Zone identified by the immediately preceding ZONID. The MATID can be either the Hollerith Material name or the integer Material number (see MATLS Array).

CONC₁, CONC₂, etc., are the basic concentrations (densities, volume fractions, etc.) for the Materials identified by the immediately preceding MATID's.

NOTE 3: EACH ZONE'S SPECIFICATION MUST BE SEPARATED FROM THE NEXT BY A SEMICOLON (;).

NOTE 4: THERE MUST BE EXACTLY NZONE ZONES SPECIFIED.

FORM 2 SPECIAL FEATURE: The Material-to-Zone specification string characterized above by ZONID, MATID₁, CONC₁, MATID₂, CONC₂ ... ; can be input in the abbreviated form

ZONID MATID ;

if the Zone identified by ZONID consists only of the single Material identified by MATID, with a concentration of 1.0.

BLOCK IV, continued

THE ASGMOD ARRAY {Required for Concentration Search, Optional Otherwise}

In order to understand the use of the ASGMOD Array, it is necessary to understand the manner in which the code uses the Material-to-Zone assignment information.

Consider a Zone, Z, containing one or more Materials, M, each of which is characterized by a cross-section set $\sigma_{x,g}(M)$, where x denotes a cross-section type (fission, absorption, etc.) and g denotes an energy group. The corresponding macroscopic cross section for the Zone, $\Sigma_{x,g}^Z$, is generated in the code by the algorithm

$$\Sigma_{x,g}^Z = \sum_{M \in Z} [CO(M,Z) + C1(M,Z)*CMOD]* \sigma_{x,g}(M)$$

where the values of CO(M,Z) are the CONC entries in the ASSIGN array (Form 2), and represent the basic concentrations for the Materials in a given Zone. The values of C1(M,Z) are concentration factors supplied by the user through the ASGMOD array described below and which are multiplied by the concentration modifier, CMOD, where

CMOD = the eigenvalue for a CONCENTRATION SEARCH

CMOD = user supplied value (see below) for ALL OTHER PROBLEMS.

The form for entering the ASGMOD array is as follows:

ASGMOD= ZONID_a MATID₁ C1₁, MATID₂ C1₂ ... ; ZONID_b MATID₃ C1₃ MATID₄
C1₄ ... ; etc.

This form is identical to that for the ASSIGN array (except for C1 concentration factors in lieu of the basic concentrations, CONC) and will not be further amplified.

NOTE 1: ALL ZONID's and MATID's used in the ASGMOD array should be of the same form (Hollerith name or integer) as those used in the ASSIGN array.

NOTE 2: The ZONID's and MATID's used in the ASGMOD array must be a subset of those used in the ASSIGN array, i.e., any MATID used in the ASGMOD array, for a given ZONID must also be used in the ASSIGN array but not conversely. Default values for the C1's for Materials used in the ASSIGN array but not in the ASGMOD array are 0.0.

BLOCK IV, continued

OTHER BLOCK IV INPUT

{Optional}

ARRAY SPECIFICATION

Name [Length]
{Present if}

ARRAY CONTENTS AND DESCRIPTIVE COMMENTS

CMOD [1]
{Optional}

The value of the concentration modifier used in the general algorithm for assigning Materials-to-Zones (see discussion under ASGMOD array in BLOCK IV). Only used if (i) ASGMOD array is input and (ii) problem being executed is not a CONCENTRATION SEARCH.

MATNAM [MT]
{Optional}

Hollerith Material Names for Materials. Used only if the MATID's used in the MATLS array were integers. First entry in MATNAM array is Hollerith name for Material number 1, second entry is Hollerith name for Material number 2, etc.

ZONNAM [NZONE]
{Optional}

Hollerith Zone Names for Zones. Used only if the ZONID entries in the ASSIGN array (and ASGMOD array, if used) were integers. First entry in ZONNAM array is Hollerith name for Zone number 1, second entry is the Hollerith name for Zone number 2, etc.

BLOCK V: SOLVER INPUT

{Required if Executing SOLVER Module}

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS		
Name	Number	[Length]	Word	Name	Comments
	{Present if}				
SOLIN	-	[5]			Basic SOLVER input control words
	{Always}		1.	IEVT	Type of calculation: -1/0/1/2/3/4 = Inhomo. source with fission or up-/scatter/Inhomo source alone/ k_{eff}/α (time absorption) search/concentration search/dimension search
			2.	ISCT	Legendre order of scattering
			3.	ITH	Mode of calculation: 0/1 = Direct forward)/Adjoint (Default = 0)
			4.	IBL	Left boundary condition: 0/1/2/3 = Vacuum/reflective*/periodic/white* {Used only for planar geometry (IGEOM = 1)}
			5.	IBR	Right boundary condition: Same options as for IBL {Used for all geometries}
					*Albedoes may be used with either reflective or white boundary conditions (see LBEDO, RBEDO arrays below)
ITER	-	[6]			Iteration and convergence control parameters
	{Optional}		1.	EPSI	Inner iteration convergence criterion (Default = 0.0001)
			2.	EPSO	Outer iteration convergence criterion (Default = EPSI)
			3.	IITL	Max number of transport inner iterations per group until $ 1.0-\lambda < 3* EPSO$ (Default chosen by code). Recommended to use default.
			4.	IITM	Max. number of transport inner iterations per group after $ 1.0-\lambda < 3* EPSO$ (Default chosen by code). Recommended to use default.
			5.	OITM	Max. number of outer iterations allowed (Default = 20)
			6.	ITLIM	Iteration time limit in seconds (Default = no limit)

BLOCK V, continued

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS		
Name	Number	[Length]	Word	Name	Comments
	{Present if}				
MISC	-	[7]			Miscellaneous control parameters
	{Optional}		1.	BHGT	Buckling height (in cm. if macroscopic cross sections in cm^{-1}). Used only for plane, cylindrical and two-angle plane geometries. {Default = 0.0 \rightarrow ∞ }
			2.	BWTH	Buckling width. Used only for plane and two-angle plane geometries. {Default = 0.0 \rightarrow ∞ }
			3.	NORM	Normalization factor: 0/NORM = No normalization/Normalize fission source rate (IEVT \geq 1) or inhomogeneous source rate (IEVT < 1) to NORM. (Integral of source rate over all angle, space, energy = NORM.) {Default = 0}
			4.	INFLUX	Input flux guess: 0/1 = None/RTFLUX standard interface file. {Default = 0}
			5.	INSORS	Input inhomogeneous source input from FIXSRC standard interface file? 0/1 = No/Yes {Default = 0}
			6.	IQUAD	Quadrature constants: -3/1/2/3/4 = SNCONS std. interface file / Built-in P_N / Built-in DP_N / card input via WGT_N , MU arrays / Built-in GQ_N . {Default = 1, but card input will override}
			7.	I2ANG	Do two-angle plane calculation? 0/1=No/Yes {Default=0} (For IGEOM=1 only.)
SOLOUT	-	[8]			Solver output control parameters
	{Optional}		1.	FLUXP	Final flux print: 0/1/2 = None/Isotropic component (scalar flux) only/All flux moments (Default = 0)
			2.	XSECTP	Macroscopic Zone cross sections print: 0/1/2 = None/Principal cross sections/All (principal plus scattering matrices). (Default = 0)
			3.	FISSRP	Final fission source rate density print: 0/1 = No/Yes (Default = 0)
			4.	SOURCP	Normalized inhomogeneous source print: 0/1 = No/Yes (Default = 0)

BLOCK V, continued

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS
Name	Number	[Length]	
		{Present if}	
			5. GEOMP Fine-mesh geometry print: 0/1 = No/Yes (Default = 0)
			6. ANGP Final angular flux print: 0/1 = No/Yes (Default = 0)
			7. RAFLUX Write angular fluxes to standard interface file RAFLUX (AAFLUX if adjoint)? 0/1 = No/Yes (Default = 0)
			8. BALP Print balance tables by coarse mesh: 0/1 = No/Yes (currently inoperative)
CHI	20*	[NGROUP;M] Fission spectrum by energy group and Zone. Data is entered as M strings each NGROUP data words long. M is any integer such that $1 < M <= NZONE$. If $M < NZONE$, the last string entered, i.e., the last NGROUP values of CHI will be used for Zones M+1, M+2, ..., NZONE. The SEMICOLON (;) STRING DELIMITER MUST BE USED BETWEEN STRINGS!
	{Required}		
	Unless IEVT= 0		
DEN	21*	[IT] Fine spatial mesh density factors to be applied to the Zone macroscopic cross sections at each mesh point. (IT= no. of fine spatial mesh intervals, see BLOCK I.)
	{Optional}		
WGT	22*	[MM] S_N quadrature weights. For Plane and Spherical Geometry $MM = ISN$ (see BLOCK I); for cylindrical geometry $MM = ISN(ISN+2)/4$; for 2-angle plane geometry $MM = ISN*(ISN+2)$. For ordering of weights, see Ch. V.
	{Optional}		
MU	23*	[MM] S_N quadrature cosines. For ordering of cosines, see Ch. V.
	{Optional}		
LBEDO	30*	[NGROUP] Left boundary albedoes for each group for plane geometry (IGEOM = 1) only. Applied as albedoes for either reflective (IBL = 1) or white (IBL = 3) boundary conditions. (Default = 1.0)
	{Optional}		
RBEDO	31*	[NGROUP] Right boundary albedoes for each group for all geometries. Applied as albedoes for either reflective (IBR = 1) or white (IBR = 3) boundary conditions. (Default = 1.0)
	{Optional}		

BLOCK V, continued

INHOMOGENEOUS (FIXED) SOURCES, (particles per unit time per unit volume)

{Required if IEVT \leq 0 AND INSORS = 0 AND NO BOUNDARY (SURFACE) SOURCES}

<u>ARRAY SPECIFICATION</u>			<u>ARRAY CONTENTS AND DESCRIPTIVE COMMENTS</u>
<u>Name</u>	<u>Number</u>	<u>[Length]</u>	
	{Present if}		
SOURCE ,	40* ,	[NGROUP;N]....	Inhomogeneous source <u>energy</u> spectra. The number, N, of strings is the number of angular source moments desired. Each string must contain NGROUP data entries. First string contains the spectrum for the isotropic component of the source. Succeeding strings contain spectra for successively higher source moments (see Ch. V for ordering of higher moments). THE SEMICOLON (;) STRING DELIMITER MUST BE USED BETWEEN STRINGS.
SOURCX ,	41* ,	[IT;N]	Inhomogeneous source <u>spatial</u> distributions. The number, N, of strings is the number of angular source moments desired. Each string must contain IT (number of fine spatial mesh intervals) data entries. First string contains the spatial distribution for the isotropic component of the source. Succeeding strings contain the distributions for successively higher moments (see Ch. V for ordering of moments). THE SEMICOLON (;) STRING DELIMITER MUST BE USED BETWEEN STRINGS.
NOTE on use of SOURCE/SOURCX:			Inhomogeneous source used by the code is formed by taking the product of energy spectrum times the spatial distribution for each moment. If one of the two arrays, SOURCE or SOURCX, is input and the other omitted, the omitted array is defaulted to unity.
SOURCEF ,	44* ,	[IT;M]	<u>Full</u> inhomogeneous source space-energy distribution. The number of strings, M, is equal to N*NGROUP, where N is the desired number of source angular moments. Each string contains IT data entries. First string is group 1, spatial distribution for isotropic component, second string is group 2 spatial distribution for isotropic component, etc. The NGROUP+1 string (if present) is the group 1 spatial distribution for the 1st angular moment, etc. STRING DELIMITERS (;) <u>MUST</u> BE USED BETWEEN STRINGS.

BLOCK V (cont.)

BOUNDARY (SURFACE) SOURCES, (particles per unit time per unit area)

{Optional}

<u>ARRAY SPECIFICATION</u>			<u>ARRAY CONTENTS AND DESCRIPTIVE COMMENTS</u>
<u>Name</u>	<u>Number</u>	<u>[Length]</u>	
SILEFT ,	50*	[NGROUP].....	Fixed, <u>isotropic</u> boundary (surface) source at the left boundary for each energy group. FOR PLANE GEOMETRY ONLY!
SALEFT ,	51*	[K;NGROUP]....	Fixed, <u>angular</u> boundary (surface) source at the left boundary for each inward-directed direction and each energy group. FOR PLANE GEOMETRY ONLY! K = ISN/2 for standard plane geometry K = ISN*(ISN+2)/2 for 2-angle plane geometry calculation (I2ANG = 1). Entered as NGROUP strings of data, each string containing K data entries, beginning with group 1. The ordering of the angular boundary sources (fluxes) is described in Ch. V.F.2. STRING DELIMITER (;) MUST BE USED BETWEEN STRINGS!
SIRITE ,	52*	[NGROUP].....	Fixed, <u>isotropic</u> boundary (surface) source at the right boundary for each energy group beginning with group 1.
SARITE ,	53*	[K;NGROUP]....	Fixed, <u>angular</u> boundary source at the right boundary for each inward-directed angular direction and each energy group. K = ISN/2 for standard plane and spherical geometries (IGEOM = 1,3); K = ISN*(ISN+2)/8 for cylindrical geometry (IGEOM = 2); K = ISN*(ISN+2)/2 for 2-angle plane geometry calculation (I2ANG = 1 and IGEOM = 1). Entered as NGROUP strings of data, each string containing K data entries, beginning with group 1. The ordering of the angular boundary sources (fluxes) is described in Ch. V.F.2. STRING DELIMITER (;) MUST BE USED BETWEEN STRINGS!

BLOCK V, continued

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS		
Name	Number	[Length]	Word	Name	Comments
		{Present if}			
SEARCH	-	[8]		Control parameters for searches and time-absorption (α) calculation
		{Always if IEVT \geq 2}	1.	IPVT	Parametric value type: 0/1/2 = None/ k_{eff}/α . IPVT = 1 or 2 only valid for IEVT \geq 2. (Default = 1 if IEVT \geq 2)
			2.	PV	Parametric value of k_{eff} (if IPVT = 1) or α (if IPVT = 2). PV is a fixed quantity during a search. (Default = 1.0 if IPVT = 1, 0.0 if IPVT = 2)
			3.	EV	Value of eigenvalue at which search is started (used only if IEVT \geq 2) (Default = 0.0)
			4.	EVM	Eigenvalue modifier to adjust eigenvalue at which search is started (used only if IEVT \geq 2). REQUIRED INPUT if IEVT $>$ 2!
			5.	XLAL	Lambda lower limit for eigenvalue searches. (Default = 0.01)
			6.	XLAH	Lambda upper limit for eigenvalue searches. (Default = 0.5)
			7.	XLAX	Lambda convergence criterion for second and subsequent search steps, i.e., search convergence criterion. (Default = 10^*EPSI)
			8.	POD	Parameter oscillation damper. See Ch. V. (Default 1.0)
RM	27*	[IM]		Radius modifiers for each mesh interval for use with dimension searches only. For coarse-mesh interval K, the coarse-mesh spatial boundary \tilde{R}_{K+1} is computed as
		{Always if IEVT = 4}			$\tilde{R}_{K+1} = \tilde{R}_K + (R_{K+1} - R_K) * (1 + EV * RM_K)$ $k = 1, \dots, IM$
					where EV is the dimension search eigenvalue being sought and R_K, R_{K+1} are the input coarse-mesh boundaries supplied in the XMESH array of BLOCK II.

BLOCK VI: EDITS

{Required if Executing EDIT Module}

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS		
Name	Number	[Length]	Word	Name	Comments
PTED	- {Always}	[1]		Do fine space-point edits: 0/1 = No/Yes
ZNED	- {Always}	[1]		Do EDIT ZONE edits: 0/1 = No/Yes
POINTS,	80\$, {Optional}	[<IT]		Fine-mesh point (or interval) numbers for which point edits are desired. Must be in ascending order. USED ONLY IF PTED = 1. (Default = all points)
EDZONE,	83\$, {Optional}	[IT]		EDIT ZONE number for each fine spatial mesh interval. USED ONLY IF ZNED = 1. (Default = SOLVER coarse mesh interval numbers, see XMesh array, BLOCK II)
ICOLL,	84\$, {Optional}	[NBG]		<p>Edit Energy-Broad-Group collapsing option. Number of Solver energy groups in each Edit Energy-Broad-Group. NBG is the number of entries in the ICOLL array. It is required that</p> $\sum_{N=1}^{NBG} ICOLL(N) = NGROUP.$ <p>(Default = 1 energy group per Edit Energy-Broad-Group)</p>
IGRPED	- {Optional}	[1]		Print option on energy groups: 0/1/2/3 = Print energy group total only /Print only Edit Energy-Broad-Groups/same as 1/Print both Edit Energy-Broad-Groups and group totals (Default = 0)
BYVOLP	- {Optional}	[1]		Print option on fine space-point edits: Printed point reaction rates will have been multiplied by the mesh interval volume? 0/1 = No/Yes. (Default = 0)
AJED	- {Optional}	[1]		0/1 = Regular (forward) Edit/Adjoint Edit. Regular edit uses RTFLUX scalar flux file; Adjoint edit uses ATFLUX scalar flux file. (Default = 0)

REACTION RATE/RESPONSE FUNCTION EDIT SPECIFICATIONS

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS
Name	Number	[Length]	Comments
	{Present if}		
EDXS	- {Optional}	[<NEDT].....	Cross section types to be used in forming reaction rates. May be entered by integer (denoting edit position of desired cross-section type) or by Hollerith name of cross-section type. See Table II. NEDT is the total number of Edit cross-section types available from the input cross section library (see Table II). (Default = All) NOTE: The cross-section types specified in this array apply to any or all of the following specified edit forms: RESDNT, EDISCS, EDCONS, EDMATS.
RESDNT	- {Optional}	[1].....	Do edits using the Resident Macroscopic cross sections: 0/1=No/Yes. By resident is meant that which actually exists at each mesh point as used in the SOLVER Module. Note: If density factors were used in SOLVER, the same density factors should be provided in the XDF array for the EDIT Module.
EDISOS	- {Optional}	[<NISO].....	Isotope identifiers for isotopes to be used in forming Isotope microscopic reaction rates. Identifiers may be entered as either (i) integers in which case an integer entry of I refers to the Ith Isotope on the basic input library, or (ii) Hollerith names of the desired Isotopes. (Default = none)
EDCONS	- {Optional}	[<NISO].....	Isotope identifiers for Isotopes in forming Resident Constituent (partial macroscopic) reaction rates. Identifiers may be entered as integers (denoting the ordered position of the Isotope on the basic input library) or as Hollerith names of the desired Isotopes. (Default = none)
EDMATS	- {Optional}	[<MT].....	Material identifiers for Materials to be used in forming Material (macroscopic) reaction-rate edits. Identifiers may either be integers (Material numbers) or Hollerith Material names. (Default = none)
RSFNAM	- {Optional}	[M].....	Hollerith names for the user-input response functions specified below. M is arbitrary but must be < 500. (Default = RSFP1, RSFP2,... , RSFPM)

TABLE II
EDIT CROSS-SECTION TYPES BY POSITION AND NAME

CROSS-SECTION INPUT VIA ISOTXS/GRUPXS			CROSS-SECTION INPUT VIA BCD CARD-IMAGES		
Type	EDIT Position	Name ^a	Type	EDIT Position	Name ^a
χ	1	CHI [^] [^] [^]	Not Used	1	CHI [^] [^] [^]
$v\sigma_f$	2	NUSIGF	$v\sigma_f$	2	NUSIGF
σ_t	3	T \emptyset TAL [^]	σ_t	3	TOTAL [^]
σ_a	4	ABS [^] [^] [^]	σ_a	4	ABS [^] [^] [^]
$\sigma_{n,p}$	5	N-PR \emptyset T	1 ^b	5	EDIT1 [^] ^c
$\sigma_{n,D}$	6	N-DEUT	2 ^b	6	EDIT2 [^] ^c
$\sigma_{n,T}$	7	N-TRIT	.		
$\sigma_{n,\alpha}$	8	N-ALPH	.		
$\sigma_{n,2n}$	9	N-2N [^] [^]	.		
$\sigma_{n,\gamma}$	10	N-GAMM	N=(IHT-3) ^b	4+N	EDITN [^] ^c
σ_f	11	N-FISS			
σ_{tr}	12	TRNSPT			

^a. Names are six character Hollerith. [^] denotes blank.

^b. Denotes position (row) in the cross-section table. All cross sections in rows (positions) 1 through IHT-3 in the cross-section library are EDIT cross sections chosen by the user.

^c. These are the default names that may be overridden with the user-option names in the EDNAMES array of BLOCK III

ARRAY SPECIFICATION			ARRAY CONTENTS AND DESCRIPTIVE COMMENTS
Name	Number	[Length]	Comments
RSFE,	85*,	[NGROUP;M]	...Response function energy distribution for each of the M different response functions desired. The number of different response functions is arbitrary (but must be fewer than 500). Data are entered as M strings each of NGROUP entries beginning with group 1. THE SEMICOLON (;) STRING DELIMITER SHOULD BE USED.
{Required if user-input response functions desired}			
RSFX,	86*,	[IT;M].....	...Response function spatial distribution by fine mesh for each of the M different functions desired ($1 \leq M < 500$). Data are entered as M strings, each of IT entries beginning with mesh point 1. THE SEMICOLON STRING DELIMITER SHOULD BE USED. (Default = 1.0 for all entries if RSFE array is entered.) NOTE: Mth response function at space point I, energy group G is computed as RSFX(I,M)*RSFE(G,M)
{Optional}			
XDF,	90*,	[IT].....	...Fine spatial mesh density factors used to multiply resident Constituent (See EDCONS), Material macroscopic (See EDMATS) and Resident Macroscopic (See RESDNT)reaction rates <u>only</u> . (Default = all values unity)
{Optional}			
MICSUM	-	[< 500 sums]...	...Cross-section reaction rate summing specifications. The MICSUM array is a packed array with data entered as follows: A set of Isotope numbers or names is given, followed by a set of cross-section type position numbers or names (see Table II). These sets are delimited with an entry of 0 (zero). Reaction rates are calculated for each Isotope specified for each cross-section type specified and summed to form the first sum. The next two sets of data are used to form the second sum, etc. Up to 500 sums can be specified. (See VII.D.1.)
{Optional}			
IRSUMS	-	[< 500 sums]...	...Response function reaction rate summing specifications. The IRSUMS array is input as follows: A set of response function numbers or names is entered and the set delimited with an entry of 0 (zero). Reaction rates are calculated using these response functions, and the rates are summed to form the first sum. The next set of data is used to form the second sum, etc. Up to 500 sums can be specified. (See VII.D.2.)
{Optional, but used only if RSFE array is input}			

V. DETAILS RELATED TO INPUT FOR ONEDANT

This chapter provides a brief sketch of the development of the multigroup, discrete-ordinates, diamond-differenced form of the Boltzmann transport equation followed by information needed by the user to understand some of the ONEDANT SOLVER module options and to correctly prepare input for the code. Specific input details unique to the EDIT module are provided separately in Ch. VII.

A. Development of the Multigroup, Discrete-Ordinates Form of the Transport Equation

The time-independent inhomogeneous Boltzmann transport equation in one space dimension is

$$\begin{aligned} \nabla \cdot \underline{\Omega} \psi(r, E, \underline{\Omega}) + \sigma(r, E) \psi(r, E, \underline{\Omega}) = & \iint dE' d\Omega' \sigma_s(r, E' \rightarrow E, \underline{\Omega} \cdot \underline{\Omega}') \psi(r, E', \underline{\Omega}') \\ & + \frac{1}{4\pi} \iint dE' d\Omega' \chi(r, E' \rightarrow E) v \sigma_f(r, E') \psi(r, E', \underline{\Omega}') + Q(r, E, \underline{\Omega}), \end{aligned} \quad (11)$$

where $\psi(r, E, \underline{\Omega})$ is the particle flux (particle number density times the particle speed) defined such that $\psi(r, E, \underline{\Omega}) dE d\underline{r} d\Omega$ is the flux of particles in the energy range dE about E , in the volume element $d\underline{r}$ about r , with directions of motion in the solid angle element $d\Omega$ about $\underline{\Omega}$. Similarly, $Q(r, E, \underline{\Omega}) dE d\underline{r} d\Omega$ is the rate at which particles are produced in the same element of phase space from sources that are independent of the flux ψ . The macroscopic total cross section is σ , the macroscopic scattering transfer probability, from energy E' to energy E through a scattering angle $\underline{\Omega} \cdot \underline{\Omega}'$, is σ_s , and the macroscopic fission cross section is σ_f . All of the quantities may be spatially dependent. The number of particles emitted isotropically ($\frac{1}{4\pi}$) per fission is v , and the fraction of these particles appearing in energy dE about E from fissions in dE' about E' is $\chi(r, E' \rightarrow E)$.

The homogeneous transport equation is the same as Eq. (11) except that Q is zero and the term representing the fission source is divided by k_{eff} . The inhomogeneous problem is referred to as a source problem and the homogeneous problem will be referred to as an eigenvalue problem. ONEDANT will solve both types of problems.

TABLE III

FORMS OF $\nabla \cdot \underline{\Omega}\psi$

<u>Geometry</u>	<u>Dependence of</u>	<u>Definition of Variables</u>	<u>$\nabla \cdot \underline{\Omega}\psi$</u>
Plane	$\psi(x, \mu)$	$\mu = \hat{e}_x \cdot \underline{\Omega}$	$\mu \frac{\partial \psi}{\partial x}$
	or	$\xi = (1 - \mu^2)^{1/2} \cos \phi$	
	$\psi(x, \mu, \phi)$	$\eta = (1 - \mu^2)^{1/2} \sin \phi$	
Cylindrical	$\psi(r, \mu, \eta)$	$\mu = \hat{e}_r \cdot \underline{\Omega}$	$\frac{\mu}{r} \frac{\partial(r\psi)}{\partial r} - \frac{1}{r} \frac{\partial(\eta\psi)}{\partial \phi}$
		$\xi = \hat{e}_z \cdot \underline{\Omega}$	
		$\eta = (1 - \xi^2)^{1/2} \sin \phi$	
		$\mu = (1 - \xi^2)^{1/2} \cos \phi$	
Spherical	$\psi(r, \mu)$	$\mu = \hat{e}_r \cdot \underline{\Omega}$	$\frac{\mu}{r^2} \frac{\partial(r^2 \psi)}{\partial r}$
			$+ \frac{1}{r} \frac{\partial[(1 - \mu^2)\psi]}{\partial \mu}$

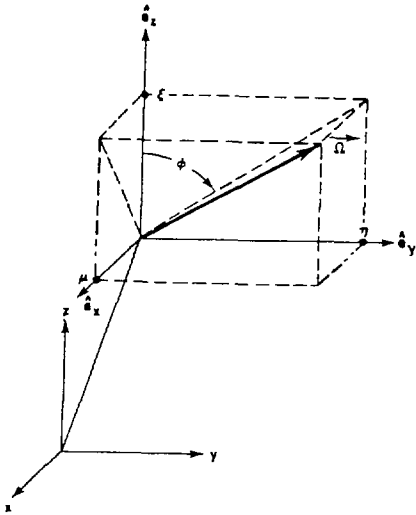


Fig. 4. Coordinates in plane geometry.

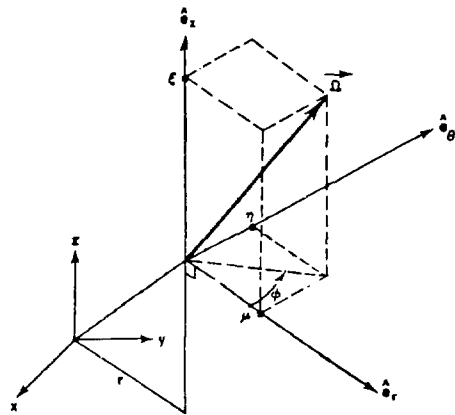


Fig. 5. Coordinates in cylindrical geometry.

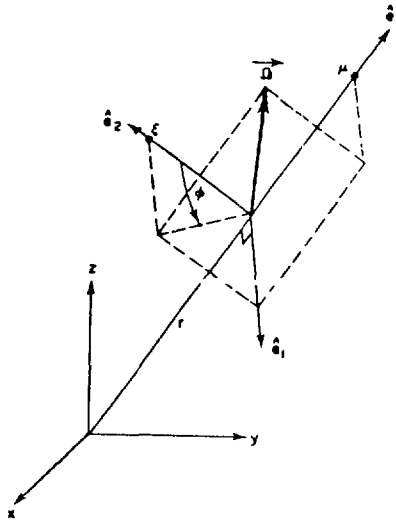


Fig. 6. Coordinates in spherical geometry.

1. Particular Forms of the Divergence Operator. The form of the divergence operator $\nabla \cdot \underline{\Omega} \psi$ (or $\underline{\Omega} \cdot \nabla \psi$) for the geometries treated by ONEDANT is given in Table 11 in terms of the coordinate systems shown in Figs. 4-6.

In the standard plane geometry, the angular flux $\psi(r, E, \underline{\Omega})$ is assumed independent of the azimuthal angle ϕ so that the angular dependence is reduced to the μ interval $(-1, +1)$. ONEDANT also permits the two-angle plane geometry option in which no assumptions of symmetry in angle are imposed. In this case the complete unit sphere of angular directions must be considered.

In cylindrical geometry, the angular flux is assumed symmetric in the ξ angular cosine and also symmetric about the μ - ξ , (or $\phi = 0^\circ - 180^\circ$) plane. Thus, only one-fourth of the unit sphere need be considered in the angular dependence.

In spherical geometry, the angular flux is assumed symmetric in the azimuthal angle ϕ so that the angular dependence is reduced to the μ interval $(-1, +1)$.

2. Spherical Harmonics Expansion of the Scattering Source. The scattering transfer probability in Eq. (11) is represented by a finite Legendre polynomial expansion of order ISCT

$$\sigma_s(r, E' \rightarrow E, \underline{\Omega} \cdot \underline{\Omega}') = \sum_{L=0}^{\text{ISCT}} \left(\frac{2L+1}{4\pi} \right) \sigma_s^L(r, E' \rightarrow E) P_n(\underline{\Omega} \cdot \underline{\Omega}') \quad . \quad (12)$$

If this expansion is inserted into Eq. (11) and the addition theorem for spherical harmonics used to expand $P_n(\underline{\Omega} \cdot \underline{\Omega}')$, the scattering source becomes

$$\begin{aligned} & \iint dE' d\Omega' \sigma_s(r, E' \rightarrow E, \underline{\Omega} \cdot \underline{\Omega}') \psi(r, E', \underline{\Omega}') \quad \equiv \text{S.S.} \\ & = \int_{E'} dE' \sum_{L=0}^{\text{ISCT}} \left(\frac{2L+1}{4\pi} \right) \sigma_s^L(r, E' \rightarrow E) \left\{ P_L(\mu) \int_{-1}^1 d\mu' \int_0^{2\pi} d\phi' P_L(\mu') \psi(r, E', \mu', \phi') \right. \\ & \quad \left. + 2 \sum_{K=1}^L \frac{(L-K)!}{(L+K)!} P_L^K(\mu) \int_{-1}^1 d\mu' \int_0^{2\pi} d\phi' P_L^K(\mu') \cos K(\phi - \phi') \psi(r, E', \mu', \phi') \right\} \end{aligned} \quad (13)$$

where for cylindrical geometry we must replace the μ variable with ξ . Using the relation $\cos L(\phi - \phi') = \cos L\phi \cos L\phi' + \sin L\phi \sin L\phi'$, we can write Eq. (13) as

$$\begin{aligned}
\text{S.S.} = & \int_{E'} dE' \sum_{L=0}^{\text{ISCT}} (2L+1) \sigma_S^L(r, E' \rightarrow E) \left\{ P_L(\mu) \Phi_L(r, E') \right. \\
& \left. + \sum_{K=1}^L \sqrt{\frac{2(L-K)!}{(L+K)!}} \left[\Phi_{C,L}^K(r, E') P_L^K(\mu) \cos K\phi + \Phi_{S,L}^K(r, E') P_L^K(\mu) \sin K\phi \right] \right\}
\end{aligned} \tag{14}$$

where we have defined the moments of the angular flux as

$$\Phi_L(r, E') \equiv \frac{1}{4\pi} \int_{-1}^1 d\mu' \int_0^{2\pi} d\phi' P_L(\mu') \psi(r, E', \mu', \phi') , \tag{15a}$$

$$\Phi_{C,L}^K(r, E') \equiv \frac{1}{4\pi} \int_{-1}^1 d\mu' \int_0^{2\pi} d\phi' \psi(r, E', \mu', \phi') P_L^K(\mu') \cos K\phi' , \tag{15b}$$

and

$$\Phi_{S,L}^K(r, E') \equiv \frac{1}{4\pi} \int_{-1}^1 d\mu' \int_0^{2\pi} d\phi' \psi(r, E', \mu', \phi') P_L^K(\mu') \sin K\phi' . \tag{15c}$$

In both standard plane and spherical geometries, due to symmetry in the azimuthal angle ϕ , the flux moments $\Phi_{C,L}^K$ and $\Phi_{S,L}^K$ are identically zero. In cylindrical geometry (with ξ, ξ' replacing μ, μ' in Eqs. (14) and (15)), the odd moments ($K+L = \text{odd}$) of $\Phi_{C,L}^K$ vanish as do all the sine moments $\Phi_{S,L}^K$. In the two-angle plane geometry all moments must be retained.

In all cases the scattering source, SS, can be written in the general form

$$SS = \int_{E'} dE' \sum_{n=1}^{NM} (2n - 1) \sigma_S^n(r, E' \rightarrow E) R_n(\underline{\Omega}) \tilde{\Phi}_n(r, E') \quad (16)$$

where NM is the total number of spherical harmonics (and flux moments) required for a given Legendre expansion order, ISCT (as shown in Table IV), the $R_n(\underline{\Omega})$ are the spherical harmonics appropriate to the particular geometry, and the $\tilde{\Phi}_n(r, E)$ are the angular flux moments corresponding to the $R_n(\underline{\Omega})$. The $R_n(\underline{\Omega})$ are listed in Table V for typical Legendre expansion orders. For each $R_n(\underline{\Omega})$ in the table is a corresponding flux moment defined by Eq. (15a), (15b), or (15c) as appropriate.

3. Spherical Harmonics Expansion of the Inhomogeneous Source. In a manner similar to that used for the scattering source, the inhomogeneous source $Q(r, E, \underline{\Omega})$ can be represented as a finite expansion using the spherical harmonics $R_n(\underline{\Omega})$ defined in Table IV. First, the inhomogeneous source moments are defined for a Legendre expansion order IQAN:

$$Q_L(r, E) \equiv \frac{1}{4\pi} \int_{-1}^1 d\mu \int_0^{2\pi} d\phi Q(r, E, \underline{\Omega}) P_L(\mu) \quad , \quad L = 0, \dots, IQAN \quad (17a)$$

$$Q_{C,L}^K(r, E) \equiv \frac{1}{4\pi} \int_{-1}^1 d\mu \int_0^{2\pi} d\phi Q(r, E, \underline{\Omega}) P_L^K(\mu) \cos K\phi \quad , \quad \begin{array}{l} L = 0, \dots, IQAN \\ K = 1, \dots, L \end{array} \quad (17b)$$

$$Q_{S,L}^K(r, E) \equiv \frac{1}{4\pi} \int_{-1}^1 d\mu \int_0^{2\pi} d\phi Q(r, E, \underline{\Omega}) P_L^K(\mu) \sin K\phi \quad . \quad (17c)$$

TABLE IV
 NUMBER OF SPHERICAL HARMONICS, N, AS A FUNCTION
 OF LEGENDRE EXPANSION ORDER, L_0

L_0	N (see below)		
	Standard Plane and Spherical Geometries	Cylindrical Geometry	Two-angle Plane Geometry
0	1	1	1
1	2	2	4
2	3	4	9
3	4	6	16
4	5	9	25
5	6	12	36

$$N = \begin{cases} L_0 + 1 & \text{for standard plane and spherical geometry} \\ (L_0 + 2)^2/4 & \text{for cylindrical geometry} \\ (L_0 + 1)^2 & \text{for two-angle plane geometry} \end{cases}$$

TABLE V
 SPHERICAL HARMONICS, $R_n(\underline{\Omega})$, FOR DIFFERENT GEOMETRIES

N	STANDARD PLANE AND SPHERICAL GEOMETRIES P_5^a	CYLINDRICAL GEOMETRY P_4^a	TWO-ANGLE PLANE P_3^a
1	$P_0(\mu)$	$P_0(\xi)$	$P_0(\mu)$
2	$P_1(\mu)$	$P_1^1(\xi) \cos\phi$	$P_1(\mu)$
3	$P_2(\mu)$	$P_2(\xi)$	$P_1^1(\mu) \cos\phi$
4	$P_3(\mu)$	$\frac{\sqrt{3}}{6} P_2^2(\xi) \cos\phi$	$P_1^1(\mu) \sin\phi$
5	$P_4(\mu)$	$\frac{\sqrt{6}}{6} P_3^1(\xi) \cos\phi$	$P_2(\mu)$
6	$P_5(\mu)$	$\frac{\sqrt{10}}{60} P_3^3(\xi) \cos 3\phi$	$\frac{\sqrt{3}}{3} P_2^1(\mu) \cos\phi$
7		$P_4(\xi)$	$\frac{\sqrt{3}}{3} P_2^1(\mu) \sin\phi$
8		$\frac{\sqrt{5}}{30} P_4^2(\xi) \cos 2\phi$	$\frac{\sqrt{3}}{6} P_2^2(\mu) \cos 2\phi$
9		$\frac{\sqrt{35}}{840} P_4^4(\xi) \cos 4\phi$	$\frac{\sqrt{3}}{6} P_2^2(\mu) \sin 2\phi$
10			$P_3(\mu)$
11			$\frac{\sqrt{6}}{6} P_3^1(\mu) \cos\phi$
12			$\frac{\sqrt{6}}{6} P_3^1(\mu) \sin\phi$
13			$\frac{\sqrt{15}}{30} P_3^2(\mu) \cos 2\phi$
14			$\frac{\sqrt{15}}{30} P_3^2(\mu) \sin 2\phi$
15			$\frac{\sqrt{10}}{60} P_3^3(\mu) \cos 3\phi$
16			$\frac{\sqrt{10}}{60} P_3^3(\mu) \sin 3\phi$

^a P_N denotes Nth order Legendre expansion

The inhomogeneous source is represented in the general spherical harmonic expansion

$$Q(r, E, \underline{\Omega}) = \sum_{n=1}^{NMQ} (2n - 1) R_n(\underline{\Omega}) \tilde{Q}_n(r, E) \quad (18)$$

where NMQ is the total number of spherical harmonics (and source moments) required for a given Legendre expansion order, IQAN, as shown in Table IV, the $R_n(\underline{\Omega})$ are the spherical harmonics appropriate to the geometry being used, and the $\tilde{Q}_n(r, E)$ are the angular source moments corresponding to the $R_n(\underline{\Omega})$. The $R_n(\underline{\Omega})$ are listed in Table V for typical Legendre expansion orders. For each of these $R_n(\underline{\Omega})$ is a corresponding source moment defined by Eq. (17a), (17b), or (17c), as appropriate.

4. Discretization of the Energy Variable-the Multigroup Approximation.

The energy domain of interest is assumed to be partitioned into NGROUP intervals of width ΔE_g , $g = 1, 2, \dots, NGROUP$. By convention, increasing g represents decreasing energy. If Eq. (11) is integrated over ΔE_g using the spherical harmonic expansion of Eqs. (16) and (18), we get

$$\begin{aligned} \nabla \cdot \underline{\Omega} \psi_g(r, \underline{\Omega}) + \sigma_g(r) \psi_g(r, \underline{\Omega}) &= \sum_{h=1}^{NGROUP} \sum_{n=1}^{NM} (2n - 1) \sigma_{s, h \rightarrow g}^n R_n(\underline{\Omega}) \tilde{\phi}_{n, h}(r) \\ &+ \sum_{h=1}^{NGROUP} (v\sigma_f)_h \chi_{h \rightarrow g} \tilde{\phi}_{1, h}(r) + \sum_{n=1}^{NMQ} (2n - 1) R_n(\underline{\Omega}) \tilde{Q}_{n, g}(r) \end{aligned} \quad (19)$$

for $g = 1, 2, \dots, NGROUP$. Here the "group flux"

$$\psi_g(r, \underline{\Omega}) = \int_{\Delta E_g} \psi(r, E, \underline{\Omega}) dE \quad (20)$$

is no longer a distribution in energy nor an average (in energy), but is the total flux of particles in the energy interval. Because of this, energy integrals in ONEDANT are evaluated by simple sums.

To simplify the notation for the following discussion, the right-hand side of Eq. (19) (the sources due to scattering, fission, and inhomogeneous source) will be denoted by $S_g(r, \underline{\Omega})$. It is recognized that portions of S_g depend on the unknown flux ψ_g through the flux moments, but this dependence is treated by iterative procedures and, accordingly, the simplified representation results in no loss of generality. Equation (19) is thus written, with the group index omitted.

$$\nabla \cdot \underline{\Omega} \psi(r, \underline{\Omega}) + \sigma(r) \psi(r, \underline{\Omega}) = S(r, \underline{\Omega}) \quad . \quad (21)$$

5. Discrete-Ordinates Equations. In the discrete-ordinates approximation, the angular-direction domain, characterized by $\underline{\Omega}$, is discretized into a set of quadrature points each with an associated quadrature weight. Although not rigorously correct, the discrete-ordinates approximation is commonly referred to as the S_N method, and the number of quadrature directions, MM , is a function of both the S_N order and the geometry, as shown in Table VI. In discretizing the angular domain $\underline{\Omega}$, each of the quadrature points, or directions, is characterized by the subscript m and corresponds to direction $\underline{\Omega}_m$. The quadrature weight, w_m , corresponds to the differential area on the unit sphere normalized to unity, i.e., w_m is analogous to $d\Omega_m/4\pi$. In such a manner the weights, w_m , are normalized so that

$$\sum_{m=1}^{MM} w_m = 1 \quad . \quad (22)$$

The angular flux for direction m at space point r is denoted $\psi_m(r)$ and represents the average angular flux in directions $d\Omega_m$ about $\underline{\Omega}_m$ at space point r . The scalar flux at space point r , $\Phi_0(r) = \tilde{\Phi}_1(r)$, is simply

$$\phi_o(\underline{r}) = \sum_{m=1}^{MM} w_m \psi_m(r) \quad . \quad (13)$$

a. Standard Plane Geometry. For standard plane geometry (see Table III and Fig. 4) azimuthal symmetry is assumed in ϕ so that $\underline{\Omega}(\mu, \phi)$ becomes $\underline{\Omega}(\mu)$ and $d\underline{\Omega}$ becomes $2\pi d\mu$. The angular interval $\mu \in [-1, 1]$ is discretized into MM quadrature points μ_m and associated weights w_m ordered as shown in Fig. 7. Note that the weights, w_m , correspond to $d\mu_m/2$ for this geometry. The angular flux moments, given by Eq. 15a, are approximated by

$$\phi_L(x) \approx \sum_{m=1}^{MM} w_m P_L(\mu_m) \psi_m(x) \quad . \quad (24)$$

TABLE VI

NUMBER OF QUADRATURE POINTS, MM, AS A FUNCTION OF S_N ORDER, N

N	MM			
	<u>Standard Plane Geometry</u>	<u>Two-Angle Plane Geometry</u>	<u>Cylindrical Geometry</u>	<u>Spherical Geometry</u>
2	2	8	2	2
4	4	24	6	4
6	6	48	12	6
8	8	80	20	8
12	12	168	42	12
16	16	288	72	16
N	N	$N \cdot (N + 2)$	$\frac{N \cdot (N + 2)}{4}$	N

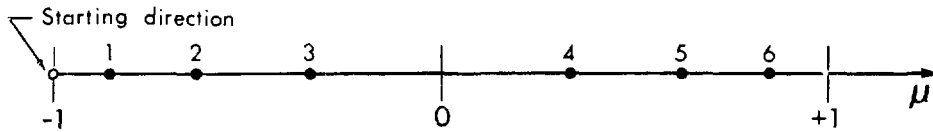


Fig. 7. Ordering of S_6 directions in plane and spherical geometries. The starting direction only applies to spherical geometry.

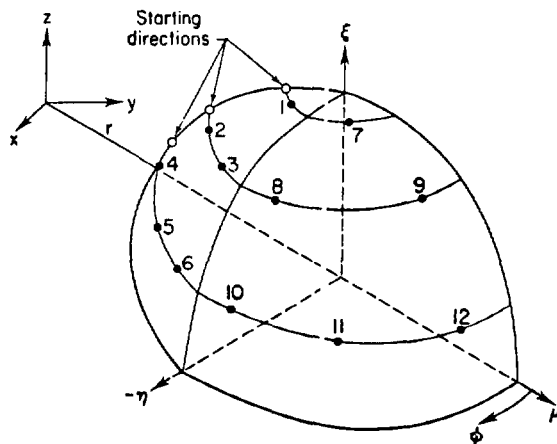


Fig. 8. Ordering of S_6 directions in cylindrical geometry.

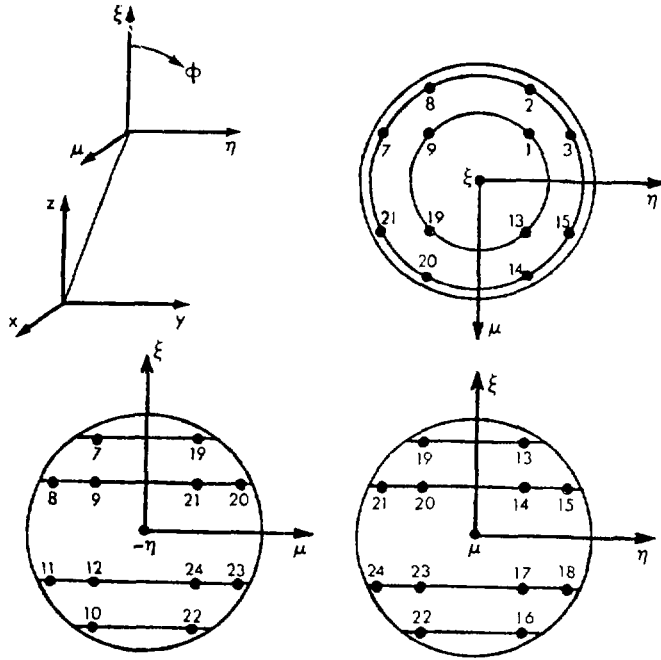


Fig. 9. Ordering of S_4 directions in two-angle plane geometry. The ordinates in the octant $\mu, \xi < 0, \eta > 0$ are not shown.

The discrete-ordinates approximation to the transport Eq. (21) becomes

$$\mu_m \frac{\partial \psi_m(x)}{\partial x} + \sigma(x) \psi_m(x) = S_m(x) \quad . \quad (25)$$

b. Two-Angle Plane Geometry. For two-angle plane geometry the entire unit sphere of directions is discretized into MM quadrature points (μ_m, ϕ_m) and associated weights ordered as shown in Fig. 9. The weights, w_m , correspond to $d\Omega_m/4\pi$ for this option. The angular flux moments, given by Eqs. (15a) - (15c), are approximated by

$$\Phi_L(x) \approx \sum_{m=1}^{MM} w_m P_L(\mu_m) \psi_m(x) \quad , \quad (26a)$$

$$\Phi_{C,L}^K(x) \approx \sum_{m=1}^{MM} w_m \psi_m(x) P_L^K(\mu_m) \cos K\phi_m, \quad (26b)$$

$$\Phi_{S,L}^K(x) = \sum_{m=1}^{MM} w_m \psi_m(x) P_L^K(\mu_m) \sin K\phi_m. \quad (26c)$$

The discrete-ordinates approximation to the transport equation is the same as for standard plane geometry, i.e., Eq. (25).

c. Cylindrical Geometry. For cylindrical geometry (see Table III and Fig. 5), the multigroup transport Eq. (21) may be written

$$\mu \frac{\partial(r\psi)}{\partial r} - \frac{\partial(\eta\psi)}{\partial \phi} + r\sigma\psi = r S(r, \underline{\Omega}) \quad (27)$$

where $\psi = \psi(r, \underline{\Omega})$.

For the discrete-ordinates approximation in cylindrical geometry, only one quadrant of the unit sphere is discretized into a set of MM quadrature points (μ_m, η_m) and associated quadrature weights w_m . The ordering of these quadrature points is illustrated in Fig. 8 for an S_6 quadrature. As before, $\psi_m(r) \hat{=} \psi(r, \mu_m, \eta_m)$ represents the average angular flux in $d\Omega_m$ about $\underline{\Omega}_m$ and the angular flux moments for direction m are given by Eqs. (26a) - (26b). In addition, it is necessary to define angular-cell-edge fluxes on a given ξ -level as $\psi_{m-\frac{1}{2}}(r)$ and $\psi_{m+\frac{1}{2}}(r)$. The discrete-ordinates approximation to Eq. (27) can then be written.

$$\mu_m \frac{\partial(r\psi_m)}{\partial r} + \left(\frac{\alpha_{m+\frac{1}{2}}}{w_m} \right) \psi_{m+\frac{1}{2}}(r) - \left(\frac{\alpha_{m-\frac{1}{2}}}{w_m} \right) \psi_{m-\frac{1}{2}}(r) + r\sigma\psi_m(r) = rS_m(r), \quad (28)$$

where the $\alpha_{m-\frac{1}{2}}$ and $\alpha_{m+\frac{1}{2}}$ are angular coupling coefficients. These coefficients satisfy the recursion relation

$$\alpha_{m+\frac{1}{2}} - \alpha_{m-\frac{1}{2}} = -w_m \mu_m \quad (29)$$

with the requirement that the first (α_1) and last ($\alpha_{M+\frac{1}{2}}$) coefficients on each ξ -level must vanish. It can be shown⁶ that Eq. (29) becomes identical to Eq. (27) in the limit of vanishingly small angular intervals.

d. Spherical Geometry. From Table III the multigroup transport Eq. (11) can be written

$$\mu \frac{\partial(r^2\psi)}{\partial r} + r \frac{\partial[(1-\mu^2)\psi]}{\partial \mu} + r^2\sigma\psi = r^2S(r,\mu) \quad (30)$$

where azimuthal symmetry in ϕ (see Fig. 6) has been assumed. The angular domain $\mu \in [-1,1]$ is discretized into M quadrature points μ_m and associated weights w_m . Note that in spherical geometry, like standard plane geometry, the w_m correspond to $d\mu_m/2$. The ordering of the quadrature points is illustrated in Fig. 7. As before, $\psi_m(r) \equiv \psi(r, \mu_m)$ represents the average angular flux in $d\Omega_m$ ($= d\mu_m$) about Ω_m and the angular flux moments, given by Eq. (15a), are approximated by Eq. (14). In addition, it is necessary to define angular-cell-edge fluxes $\psi_{m-\frac{1}{2}}(r)$ and $\psi_{m+\frac{1}{2}}(r)$. The discrete-ordinates approximation to (30) is then written as

$$\begin{aligned} \mu_m \frac{\partial(r^2\psi_m)}{\partial r} + \left[\left(\frac{\beta_{m+\frac{1}{2}}}{w_m} \right) \psi_{m+\frac{1}{2}}(r) - \left(\frac{\beta_{m-\frac{1}{2}}}{w_m} \right) \psi_{m-\frac{1}{2}}(r) \right] r \\ + r^2\sigma\psi_m(r) = r^2S_m(r) \end{aligned} \quad (31)$$

where the angular coupling coefficients β must satisfy the recursion relation

$$\beta_{m+\frac{1}{2}} - \beta_{m-\frac{1}{2}} = -2w_m \mu_m, \quad m = 1, \dots, MM, \quad (32)$$

with the requirement from particle conservation that the first ($\beta_{\frac{1}{2}}$) and last ($\beta_{MM+\frac{1}{2}}$) coefficients must vanish. It can be shown⁶ that Eq. (31) becomes identical to Eq. (30) in the limit of vanishingly small angular intervals.

e. Starting Directions. For the curved geometries discrete-ordinates Eqs. (28) and (31), there are three variables to be determined at each space position, r : the angular-cell-edge fluxes $\psi_{m-\frac{1}{2}}(r)$ and $\psi_{m+\frac{1}{2}}(r)$ and the average angular flux $\psi_m(r)$. The $\psi_{m-\frac{1}{2}}(r)$ flux can be assumed known (except for $\psi_{\frac{1}{2}}(r)$) from the previous angular mesh-cell computation and assuming continuity at the angular mesh-cell boundaries. The standard diamond-difference⁷ assumption in angle is made to relate the $\psi_{m+\frac{1}{2}}$ to ψ_m , viz.,

$$\psi_m(r) = \frac{1}{2} \left[\psi_{m-\frac{1}{2}}(r) + \psi_{m+\frac{1}{2}}(r) \right]. \quad (33)$$

Using (33) to solve for $\psi_{m+\frac{1}{2}}$ and substituting the resulting expression into (28) or (31), there remains but one equation for the one unknown $\psi_m(r)$.

The assumption that $\psi_{m-\frac{1}{2}}$ is known is correct except for $m = 1$ for which an initial, or starting, condition is required. To achieve this, ONEDANT uses special, zero-weighted starting directions in spherical and cylindrical geometries to calculate $\psi_{\frac{1}{2}}(r)$. For spherical geometry this starting direction is the straight-inward direction $\mu = -1$ for which the term $(1 - \mu^2)\psi$ in Eq. (30) vanishes. This yields a special form of Eq. (31) which can be solved for $\psi_{\frac{1}{2}}(r)$. For cylindrical geometry, as shown in Fig. 8, starting directions corresponding to ordinates directed towards the cylindrical axis, $\eta = 0$, $\phi = 180^\circ$, are used for each ξ -level to yield special equations for $\psi_{\frac{1}{2}}(r)$ on each ξ -level.

6. Discretization of the Spatial Variable. The spatial domain of the problem is ultimately partitioned into IT fine-mesh intervals of width Δx_i , $i = 1, 2, \dots, IT$ such that $\Delta x_i \equiv x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. Subscripts with half-integer values denote interval boundaries, and integer subscripts denote interval average, or midpoint, values. It is assumed that $x_{i+\frac{1}{2}} > x_i > x_{i-\frac{1}{2}}$. With such a partitioning, space derivatives are approximated by finite differences and, typically, the resulting equations are cast in forms using interval, or mesh, average fluxes, sources, etc. As an example, for group g , the average scalar flux for interval i is

$$\phi_{i,g} \equiv \sum_{m=1}^{MM} w_m \psi_{m,i,g} \quad .$$

B. Input of Geometry-Related Information

Geometry-related information is passed to the ONEDANT SOLVER and EDIT modules solely by means of a GEODST standard interface file.¹ If no GEODST file exists prior to the execution of the code package, the user may instruct the Input Module to create the desired GEODST file by (i) providing BLOCK II input data in the card-image input file, and (ii) setting (or defaulting) the BLOCK I input parameter, NOGEOD, to zero. If, on the other hand, a pre-existing GEODST file is to be used, the user may so instruct the code by either (i) omitting all BLOCK II input from the card-image input file or (ii) setting the BLOCK I input parameter NOGEOD to unity.

In the specification of geometry and space-variable related input, the user must be familiar with the nomenclature used by ONEDANT. The terms FINE MESH, COARSE MESH, and ZONES are defined below. The term REGION is not used directly by ONEDANT but is used in the GEODST standard file which ONEDANT will accept as input.

The FINE MESH is the spatial solution-mesh for the problem, as described in the preceding section. Each FINE MESH, or FINE MESH INTERVAL, is bounded by an adjacent pair of fine-mesh grid-lines $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ with $x_{i-\frac{1}{2}} < x_{i+\frac{1}{2}}$. There are IT such FINE MESH INTERVALS. No Material discontinuities may occur within a FINE MESH INTERVAL. The specification of the FINE MESH is accomplished by means of the COARSE MESH.

The COARSE MESH is a spatial superset of the FINE MESH and is formed by partitioning the spatial domain of the problem into a suitable number of "coarse" intervals. There are IM COARSE MESH INTERVALS spanning the problem. Each COARSE MESH INTERVAL is bounded by an adjacent pair of coarse-mesh boundaries that are specified in the input either as the XMESH array in BLOCK II or as the XMESH array on a GEODST standard interface file. Each COARSE MESH INTERVAL contains one or more FINE MESH INTERVALS. The number of FINE MESH INTERVALS per COARSE MESH INTERVAL is specified by means of either the XINTS array in input BLOCK II or the IFINTS array on a GEODST file. All FINE MESH INTERVALS within a COARSE MESH INTERVAL have equal widths. No material discontinuities may occur within a COARSE MESH INTERVAL.

The REGION is a spatial superset of COARSE MESH INTERVALS or, conversely, a spatial subset of a ZONE. A REGION contains one or more COARSE MESH INTERVALS and one or more REGIONS comprise a ZONE. No material discontinuities occur within a REGION. The concept of the REGION is used only in conjunction with input from a GEODST standard interface file. For input through BLOCK II card-images, the term REGION is treated synonymously with the term COARSE MESH INTERVAL.

The ZONE is a spatial superset of COARSE MESH INTERVALS and is characterized by a single set of multigroup nuclear properties, i.e., cross sections, so that all FINE MESH INTERVALS within a ZONE have the same cross sections. A ZONE number is assigned to each COARSE MESH INTERVAL by either (i) the ZONES array in input BLOCK II, or (ii) the NZNR and MR arrays on a GEODST standard file.

C. Input of the Basic Cross-Section Library

The general procedure for generating the macroscopic cross sections appropriate to each zone in the problem is to begin with a basic library containing multigroup cross-section data for Isotopes. This section describes the allowable forms that these libraries can take and gives additional details regarding the BLOCK III input.

1. ISOTXS and GRUPXS Standard Interface Files. Either of the standard interface files ISOTXS or GRUPXS¹ can be used for providing the basic, multigroup cross sections for Isotopes. ISOTXS is an Isotope-ordered, binary file while GRUPXS is a group-ordered binary file. A complete description of these standard interface files is found in Ref. 1.

If the basic library of Isotope cross sections is an ISOTXS file, the user enters LIB= ISOTXS in the BLOCK III input; if the library is a GRUPXS file, the user enters LIB= GRUPXS.

2. Card-Image, BCD Libraries. The basic multigroup cross sections for Isotopes can be provided in a card-image, BCD library. This library consists of a collection of cross-section tables. Each of these cross-section tables contains the full set of multigroup cross sections for one Legendre scattering order for one Isotope. The ordering of cross sections within a cross-section table, the ordering of cross-section tables to form the library, and other details and user options are described below.

The user specifies that the library of cross sections is to be a card-image, BCD library by entering either LIB= ODNINP or LIB= XSLIB in the BLOCK III input. If LIB= ODNINP, the library card-images are physically located within the input for the ONEDANT code between the input for BLOCK III and the input for BLOCK IV. If LIB= XSLIB, the library card-images are physically located on a file named XSLIB, which must exist at the time of code execution.

a. Ordering of Cross Sections within a Cross-Section Table. The BCD card-image library form assumes that each cross-section table in the library contains an array of cross sections of IHM rows for each of NGROUP group columns. The cross-section type for each group is determined by its row position as shown in Table VII. Row positions are specified relative to the positions of the total cross section σ_t (row IHT) and the within-group scattering cross section $\sigma_s(g \rightarrow g)$, (row IHS). Note that the values of IHM, IHT, and IHS are input values in BLOCK III.

Each cross-section table contains the cross sections for one Legendre scattering order for one Isotope as IHM*NGROUP data entries. A cross-section table begins on a new card-image and the data are entered continuously beginning with IHM entries for group 1, followed by IHM entries for group 2, etc.

b. Card-Image Data Formats. The cross-section data may be entered on the card-images in one of three data formats, the traditional Los Alamos format, the fixed-field FIDO format, or the free-field FIDO format. The user selects the desired format through the IFIDO input parameter in the BLOCK III input.

In the traditional Los Alamos format (IFIDO= 0), also called the DTF format, the data are entered on the card-images in 6E12.0 format.

In the fixed-field FIDO format (IFIDO= 1), sometimes called the ANISN format, the data are entered on the card-images using the fixed-field FIDO format described in Ch. III.C. When this format is used, each cross-section table must be terminated with the "T" terminator described in Ch. III.C.

In the free-field FIDO format (IFIDO=2), the data are entered on the card-images in free-field format as described in Ch. III.A. When this format is used, each cross-section table must be terminated with the "T" terminator described in Ch. III. A. 2 (c).

NOTE: For free- or fixed-field FIDO cross sections, neither an array name (or number) nor an array identifier is needed with the cross-section data.

c. Cross-Section Table Title Cards. A single title card may optionally be attached to the front of each cross-section table, if desired. This option is controlled by the input parameter, ITITL in the BLOCK III input.

d. Anisotropic Scattering and the Ordering of Cross-Section Tables.

In the ONEDANT code package it is assumed that the scattering transfer probability can be represented by the finite Legendre polynomial expansion of Eq. (12), which, in multigroup notation, becomes

$$\sigma_s(g' \rightarrow g, \mu_o) = \sum_{n=0}^{\text{ISCT}} \frac{2n+1}{4\pi} P_n(\mu_o) \sigma_s^n(g' \rightarrow g) \quad (34)$$

where $\mu_o \equiv \underline{\Omega}' \cdot \underline{\Omega}$, the scattering angle and ISCT is the desired Legendre order of anisotropy in the transport calculation (input in BLOCK V). If ISCT > 0, additional tables of cross sections must be supplied in order to provide the higher order scattering cross sections $\sigma_s^n(g' \rightarrow g)$ needed for the Legendre expansion.

When using the BCD card image library, the first cross-section table for an Isotope contains the P_0 , or isotropic, multigroup cross sections ordered as shown in Table VII. The next cross-section table provides the P_1 multigroup cross sections with the same ordering; the next table contains the P_2 multigroup cross sections, etc., all for the same Isotope. It should be noted that, for higher Legendre order cross-section tables, only the scattering cross sections are used. The first IHT rows for each group are ignored in the P_L ($L > 0$)

TABLE VII

CROSS-SECTION ORDERING IN CARD-IMAGE LIBRARY

	<u>Row</u>	<u>Cross-Section Type, Group g</u>	
Number of rows per group ↑ IHM* ↓	.	.	} Edit Positions, (Optional)
	.	.	
	.	.	
	IHT-2	σ_a	} Principal Cross Sections
	IHT-1	$\nu\sigma_f$	
	IHT ^a	σ_t	
	.	.	
	.	.	} Upscatter Cross Sections
	.	.	
	IHS-2	$\sigma_s(g+2 \rightarrow g)$	}
	IHS-1	$\sigma_s(g+1 \rightarrow g)$	
	IHS ^a	$\sigma_s(g \rightarrow g)$	
	IHS+1	$\sigma_s(g-1, g)$	}
	IHS+2	$\sigma_s(g-2, g)$	
	.	.	
.	.	} Downscatter Cross Sections	
.	.		
.	.		

^a Input into BLOCK 111.

tables and the data values in these positions are usually input as 0.0. The number of tables per isotope can vary with each isotope. The number of cross-section tables per isotope is provided in the input array NTPI in BLOCK III. If the NTPI array is not provided, the code will assume that the card-image library contains MAXORD + 1 cross-section tables for each Isotope, where MAXORD is an input parameter in BLOCK III.

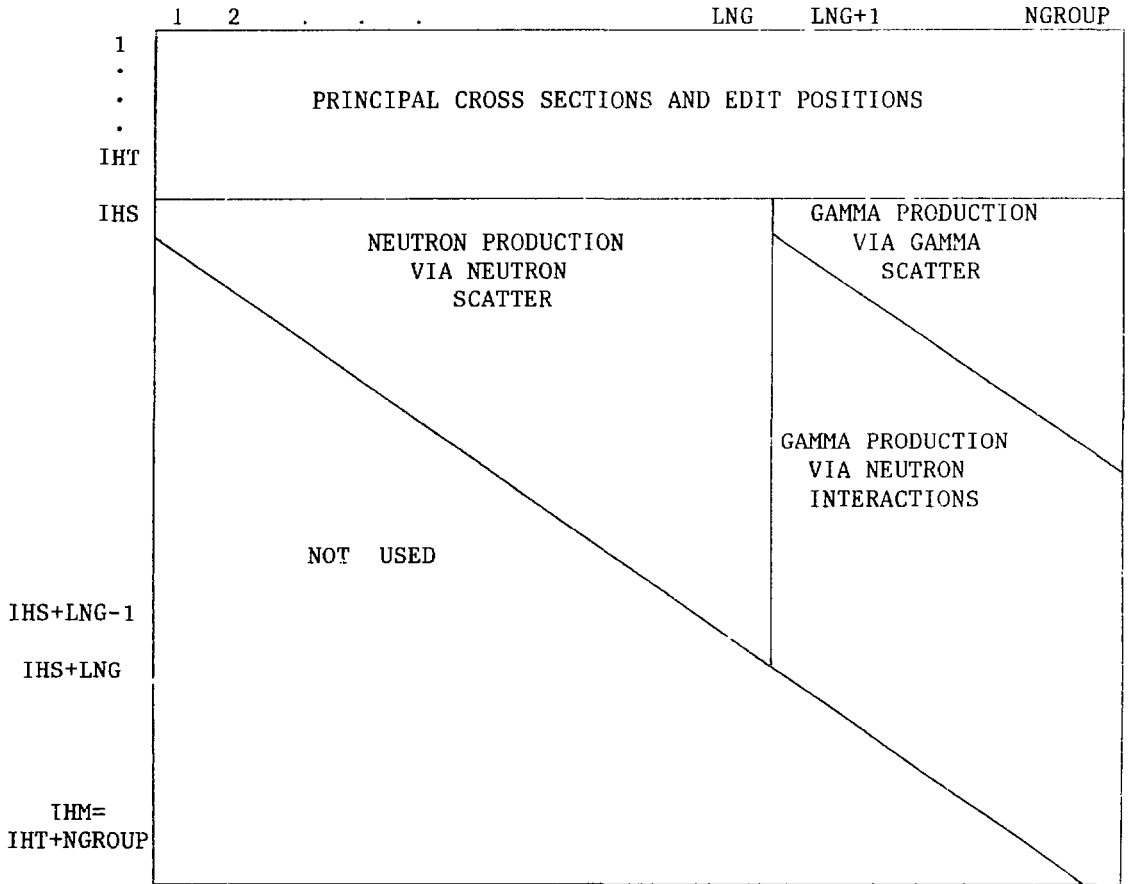
Note that the library may contain scattering data for up to a MAXORD order of anisotropy, but the actual transport calculation can be performed assuming an ISCT order of anisotropy so long as $ISCT \leq MAXORD$.

3. Binary Form of Card-Image Libraries. The processing of large, card-image, BCD libraries can be relatively time-consuming, especially if the library is in FIDO format. The binary form of the card-image library can be processed much more rapidly. By entering LIB= BXSLIB, the user can instruct the code to use the binary form of the card-image, BCD library (the binary file named BXSLIB) as the input for the basic cross-section data.

Use of LIB= BXSLIB requires that the appropriate binary form of the card-image library exists and is available to the code at the time of execution. To create the BXSLIB file, the user must make his initial execution with the BCD, card-image library (LIB= XSLIB or LIB= ODNINP) as previously described. However, by setting the input parameter SAVBXS= 1 in the BLOCK III input, the user can instruct the code to create the binary BXSLIB file and to retain this file after execution of the INPUT Module. The user can then save this BXSLIB binary file and use it for subsequent runs in place of the BCD library. It should be noted that in addition to the actual cross-section data, the BXSLIB file will contain any and all other information specified in the CARDS, NAMES, EDNAME, NTPI, VEL, and EBOUND arrays of BLOCK III as provided in the originating LIB= ODNINP or LIB= XSLIB run. The file description for the BXSLIB binary file is provided in App. A.

4. MACRXS and SNXEDT Cross-Section Files. By entering LIB= MACRXS in the BLOCK III input, the user can instruct the code to use the code-dependent interface files MACRXS and SNXEDT together with the standard interface files NDXSRF and ZNATDN¹ without referring to a basic library of multigroup Isotope cross sections. These four files contain cross sections and other information per-

TABLE VIII
 ARRANGEMENT OF DATA IN A COUPLED NEUTRON-GAMMA
 LIBRARY TABLE
 G R O U P



taining to the Materials created from the original Isotopes. (A more detailed discussion of the MACRXS and SNXEDT file preparation process is provided in Section D of this chapter.) This procedure circumvents the sometimes time-consuming process of re-creating these files when a series of code calculations are being made on the same basic problem.

If the user enters LIB= MACRXS, it is understood that the MACRXS, SNXEDT, NDXSRF, and ZNATDN files must have been previously created and saved and, further, that these files must be available to ONEDANT at the time of execution as follows:

TABLE IX

EXAMPLE CROSS-SECTION TABLE FOR A 7 GROUP COUPLED SET
WITH 4 NEUTRON, 3 GAMMA GROUPS

NGROUP = 7
LNG = 4
IHT = 3
IHS = 4
IHM = 10

	NEUTRON GROUPS				GAMMA GROUPS		
	1	2	3	4	5	6	7
1	$\sigma_a(1)$	$\sigma_a(2)$	$\sigma_a(3)$	$\sigma_a(4)$	$\sigma_a(5)$	$\sigma_a(6)$	$\sigma_a(7)$
2	$\nu\sigma_f(1)$	$\nu\sigma_f(2)$	$\nu\sigma_f(3)$	$\nu\sigma_f(4)$	0	0	0
3	$\sigma_t(1)$	$\sigma_t(2)$	$\sigma_t(3)$	$\sigma_t(4)$	$\sigma_t(5)$	$\sigma_t(6)$	$\sigma_t(7)$
4	$\sigma(1\rightarrow 1)$	$\sigma(2\rightarrow 2)$	$\sigma(3\rightarrow 3)$	$\sigma(4\rightarrow 4)$	$\sigma(5\rightarrow 5)$	$\sigma(6\rightarrow 6)$	$\sigma(7\rightarrow 7)$
5	0	$\sigma(1\rightarrow 2)$	$\sigma(2\rightarrow 3)$	$\sigma(3\rightarrow 4)$	$\sigma(4\rightarrow 5)$	$\sigma(5\rightarrow 6)$	$\sigma(6\rightarrow 7)$
6	0	0	$\sigma(1\rightarrow 3)$	$\sigma(2\rightarrow 4)$	$\sigma(3\rightarrow 5)$	$\sigma(4\rightarrow 6)$	$\sigma(5\rightarrow 7)$
7	0	0	0	$\sigma(1\rightarrow 4)$	$\sigma(2\rightarrow 5)$	$\sigma(3\rightarrow 6)$	$\sigma(4\rightarrow 7)$
8	0	0	0	0	$\sigma(1\rightarrow 5)$	$\sigma(2\rightarrow 6)$	$\sigma(3\rightarrow 7)$
9	0	0	0	0	0	$\sigma(1\rightarrow 6)$	$\sigma(2\rightarrow 7)$
10	0	0	0	0	0	0	$\sigma(1\rightarrow 7)$

- (i) MACRXS is required if the SOLVER module is to be executed, and
- (ii) SNXEDT, NDXSRF, and ZNATDN are required if the EDIT module is to be executed.

5. Coupled Neutron-Gamma Cross-Section Sets. The ONEDANT code package can solve coupled neutron-gamma problems in which neutron interactions with matter produce a source of gamma rays (photons). The simultaneous solution of the neutron-gamma transport problem can be effected by simply providing a coupled neutron-gamma cross-section library or set. In such a coupled set the gamma energy groups are treated as if they were the lowest energy neutron groups. For example, a 42-group coupled set (NGROUP = 42) might have 30 neutron groups (LNG = 30) followed by 12 gamma groups. Such coupled sets can be provided in the form of BCD, card-image libraries with no upscatter, so that IHS = IHT+1 and IHM = IHT+NGROUP. In this form neutrons appear to "downscatter" into the gamma-ray groups as a result of gamma production resulting from neutron interactions but gamma-rays do not "upscatter" into neutron groups, i.e., neutron production via photoneutron, or γ -n, reactions is not allowed. Using the BCD, card-image form of a coupled library with cross sections ordered as shown in Table VII (for no upscatter), the isotopic cross sections for each Legendre order of scatter carry data arranged as shown in Table VIII. Table IX shows the contents of a BCD, card-image cross-section table for a 7-group coupled set (4 neutron, 3 gamma groups).

D. Material Mixing and the Creation/Use of Interface Files

1. Material Mixing and the Creation of Interface Files. If "Isotopic" cross sections are provided from an ISOTXS, GRUPXS, or BCD card-image library, it is necessary to mix the Isotopes to create Materials. The mixing instructions are provided either by (i) card-image input in BLOCK IV by means of the MATLS array and, optionally, the PREMIX array, whose specifications are described in Ch. IV. C, or by (ii) the standard interface files NDXSRF and ZNATDN.^{1,2} If the NDXSRF and ZNATDN files are used, the term "Zone" in the file descriptions of Ref. 1 and 2 must be replaced with the word "Material" to be consistent with ONEDANT terminology.

In the Material mixing operation in the INPUT module of ONEDANT, the following four binary interface files are produced: MACRXS, SNXEDT, NDXSRF, and ZNATDN. These, and only these, files are used by subsequent portions of the code; the basic Isotopic cross-section library is "forgotten" once these four files are created.

The MACRXS code-dependent, binary interface file is described in Appendix A and contains Material cross sections in energy-group order. The MACRXS file is the only cross-section file available to the SOLVER Module. If a large Isotope-ordered, basic cross-section library is used, the mixing and group-ordering process used in creating the MACRXS file can be quite time-consuming. If several calculations are to be performed, e.g., parametric studies, on a particular nuclear system, it is advantageous to create a basic MACRXS material file one time ONLY and save this file for use in subsequent runs involving the SOLVER module. By use of the assignment-of-materials-to-zones specification, described in BLOCK IV of Ch. IV.C. and in section E of this chapter, a single set of Materials, i.e., a single MACRXS file can be used for calculating numerous different problems in which the problem Zone compositions consist of different proportions of materials. The manner in which the code is instructed to use an existing MACRXS file is described in V.D.2 below.

The SNXEDT code-dependent, binary interface file produced by the INPUT module contains group-ordered cross-section data for use by the EDIT module. Contained in the file are the principal cross sections and edit position data for all Isotopes on the basic input cross-section library. Scattering, or transfer, matrices are not included on the SNXEDT file. This file is used directly by the EDIT module for providing microscopic and constituent edits described in Chapter VII. The SNXEDT file description is given in App. A.

The NDXSRF and ZNATDN standard interface files are used by the EDIT module together with the SNXEDT file to mix the Isotopes into the Materials used by the SOLVER module. The EDIT module uses these Materials in providing the macroscopic (or Material) edits described in Chapter VII. It is again noted that in using the NDXSRF and ZNATDN files, the term "Zone" in the file descriptions of Ref. 1, 2 must be replaced with the word "MATERIAL" to be consistent with ONEDAN™ terminology.

As with the MACRXS file discussed above, it is frequently advantageous to save the SNXEDT, NDXSRF, and ZNATDN files created in one run for use in

subsequent runs, if possible. This procedure eliminates the need to continually repeat the often time-consuming process of re-creating the group-ordered code-dependent SNXEDT file. Parametric studies on variations of material compositions in the zones of the physical problem can be accomplished simply by changing the assignment-of-materials-to-zones specifications described in BLOCK IV of Ch. IV. C and in the next section of this chapter.

The manner in which the code is instructed to use existing SNXEDT, NDXSRF, and ZNATDN files is described below. It should be noted that, the use of an SNXEDT file by the EDIT module is usually accompanied by the use of the associated NDXSRF and ZNATDN files, and it is wise to treat these three files as a single triumvirate.

2. Using Existing MACRXS, SNXEDT, NDXSRF, ZNATDN Interface Files. If an existing pair of NDXSRF and ZNATDN standard interface files is to be used to specify the Material mixing instructions in conjunction with a basic Isotope cross-section library, the user should

- (i) omit the specification of the MATLS array in the BLOCK IV card-image input or, alternatively, set the BLOCK I input parameter NOMIX to unity, and
- (ii) ensure that the NDXSRF and ZNATDN binary files exist and are available to ONEDANT at the time of execution.

If an existing quartet of MACRXS, SNXEDT, NDXSRF, and ZNATDN binary interface files is to be used, the user should

- (i) omit BLOCK II and the MATLS array in BLOCK IV in the card-image input or, alternatively, set LIB= MACRXS in the BLOCK III input or, alternatively, set the BLOCK I input parameters NOMIX and NOMACR both in unity, and
- (ii) ensure that the MACRXS, SNXEDT, NDXSRF, and ZNATDN binary files exist and are available to ONEDANT at the time of execution. Note: only the MACRXS file is needed for execution of the SOLVER module, and only the SNXEDT, NDXSRF, and ZNATDN files are needed for execution of the EDIT module.

E. Assignment of Materials to Zones

The macroscopic cross sections for the Zones in the physical problem being analyzed are created from the Material cross sections by assigning Materials to Zones with appropriate Material concentrations, volume fractions, or densities, as desired. This assignment is accomplished either by means of the ASSIGN array card-image input in BLOCK IV or by means of a pre-existing code-dependent binary interface file ASGMAT.

As an example of the Material assignments to Zones, suppose the following Materials have been created: Stainless Steel (SS), Coolant (NA), U-238 Oxide (U802), U-235 Oxide (U502), and Pu-239 Oxide (PU902). It is desired to assign these three materials to create the correct macroscopic zone sections for the three Zones named CORE, BLKT, and REFL whose compositions are as follows:

<u>ZONE</u>	<u>Material</u>	<u>Material Volume Fraction</u>
CORE	SS	0.25
"	NA	0.40
"	U802	0.20
"	PU902	0.15
BLKT	SS	0.25
"	NA	0.40
"	U802	0.349
"	U502	0.001
REFL	SS	0.30
"	NA	0.70

The above specifications can be provided via the ASSIGN array of BLOCK IV of the input by entering the card-image input:

```
ASSIGN= CORE SS 0.25 NA 0.40 U802 0.20 PU902 0.15;  
        BLKT SS 0.25 NA 0.40 U802 0.349 U502 0.001;  
        REFL SS 0.3 NA 0.7
```

The card-image input for the assignment-of-materials-to-zones is written to a code-dependent, binary interface file named ASGMAT for use by both the SOLVER and EDIT modules. The file description for ASGMAT is given in App. A.

If it is desired to use a previously created ASGMAT file for specification of the assignment-of-materials-to-zones, the user should

- (i) omit the ASSIGN array specifications in the BLOCK IV card-image input or, alternatively, set the BLOCK I input parameter NOASG to unity, and
- (ii) ensure that the binary ASGMAT file exists and is available to ONE-DANT at the time of code execution.

F. Input of Inhomogeneous Sources

The SOLVER module of ONEDANT will solve the inhomogeneous form of the transport equation, Eq. (11), using the multigroup, discrete-ordinates approximation outlined in Section A of this chapter. The user specifies this type of calculation by setting the input control word IEVT to 0 or -1 (IEVT is found in the collective input array SOLIN in BLOCK V). IEVT=0 is used when there is no fissionable material in the problem and IEVT=-1 is used when fissionable material is present (but not in sufficient amount to make the system nuclearly critical or supercritical).

The user must supply the specifications for the inhomogeneous sources either in the input or from a FIXSRC¹ standard interface file. The inhomogeneous sources may be spatially distributed on the interior of the problem (distributed source) and/or may be external boundary (surface) sources. If the sources are to be input via a FIXSRC standard interface file, the user sets the input control word INSORS to 1 (INSORS is found in the collective input array MISC in BLOCK V). If INSORS is not input with value of unity, the user must supply the source specifications in the input of BLOCK V as described below.

1. Distributed Source Input. As described in section V.A.3, the inhomogeneous distributed source must be represented by the spherical harmonic expansion, Eq. (18), in multigroup form:

$$Q_g(r, \underline{\Omega}) = \sum_{n=1}^{NMQ} (2n-1) R_n(\underline{\Omega}) \tilde{Q}_{n,g}(r) \quad , \quad g = 1, \dots, \text{NGROUP} \quad . \quad (35)$$

Through the SOURCF or the SOURCE and/or SOURCX input arrays in BLOCK V of the input, the user inputs the $\tilde{Q}_{n,g}(r)$ of Eq. (35). If input is via the SOURCF

array, the input values are used directly as $\tilde{Q}_{n,g}(r)$. If input is via either SOURCE or SOURCX (or both) arrays, the input must be supplied such that SOURCE (g,n)* SOURCX (r,n) = $\tilde{Q}_{n,g}(r)$. The number of moments, NMQ, in Eq. (35) is determined solely from the number of moments supplied in the input arrays. The proper number of moments for a given Legendre order of anisotropy of the distributed source is shown in Table IV for each geometry. For example, if one wishes to enter a P_3 inhomogeneous source in cylindrical geometry, Table IV shows that six spherical harmonics are required for P_3 in cylindrical geometry. Table V shows that source moments for the spherical harmonics $P_0(\xi)$, $P_1^1(\xi) \cos \phi$, $P_2(\xi)$, $\frac{\sqrt{3}}{6} P_2^2(\xi) \cos 2\phi$, $\frac{\sqrt{6}}{6} (P_3^1(\xi) \cos \phi)$, and $\frac{\sqrt{10}}{60} P_3^3(\xi) \cos 3\phi$ are needed. These moments are defined by Eqs. (17a) and (17b) using multigroup notation, and recalling that for cylindrical geometry μ is replaced by ξ . The six moments to be supplied in the input are thus:

$$\tilde{Q}_{1,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi P_0(\xi) Q_g(r, \xi, \phi) = Q_{0,g}(r)$$

$$\tilde{Q}_{2,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi P_1^1(\xi) \cos \phi Q_g(r, \xi, \phi) = Q_{c,1,g}^1(r)$$

$$\tilde{Q}_{3,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi P_2(\xi) Q_g(r, \xi, \phi) = Q_{2,g}(r)$$

$$\tilde{Q}_{4,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi \frac{\sqrt{3}}{6} P_2^2(\xi) \cos 2\phi Q_g(r, \xi, \phi) = Q_{c,2,g}^2(r)$$

$$\tilde{Q}_{5,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi \frac{\sqrt{6}}{6} P_3^1(\xi) \cos\phi Q_g(r, \xi, \phi) = Q_{c,3,g}^1(r)$$

$$\text{and } \tilde{Q}_{6,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi \frac{\sqrt{10}}{60} P_3^3(\xi) \cos 3\phi Q_g(r, \xi, \phi) = Q_{c,3,g}^3(r)$$

for $n=1, \dots, \text{NGROUP}$. It should be recognized that the source moments above are not input as continuous variables in space, r , but are input by fine spatial mesh interval i , $i=1, \dots, \text{IT}$.

It is worth noting that most inhomogeneous distributed sources are assumed to be isotropic, so that NMQ in Eq. (35) is unity and the only source moment entered is the zeroth moment

$$\tilde{Q}_{1,g}(r) = \frac{1}{4\pi} \int_{-1}^1 d\xi \int_0^{2\pi} d\phi Q_g(r, \xi, \phi) = Q_{0,g}(r)$$

which, in fact, is simply the scalar source distribution.

The units on the input source moments $\tilde{Q}_{n,g}(r)$ are [particles per unit time and unit volume].

2. Surface (Boundary) Source Input. With a surface (boundary) source present, the incoming angular flux on the surface is set equal to a user-supplied source, Q_m :

$$\psi(\mu_m)_{\text{incoming}} = Q_m \quad .$$

The user-supplied source is group-dependent and may either be angularly isotropic or angle-dependent. The user-supplied sources may be input either by BLOCK V card-image input or via a FIXSRC^{1,2} standard interface file.

For card-image input the left boundary surface sources are input via the SILEFT array (angularly isotropic) or the SALEFT array (for angle-dependent sources). Similarly, right boundary surface sources are input via the SIRITE or SARITE arrays. Note that surface sources may only be input at either the left or right (or both) external boundaries of the physical problem. For the angle-dependent surface sources, only the incoming directions are required, but they must be ordered in the same order as the quadrature directions and weights (see Figs. 7,8,9).

For input of surface sources via a FIXSRC standard interface file, the user-input parameter INSORS in Block V must be set to unity and the appropriate FIXSRC file must be available to ONEDANT at the time of code execution. Note that ONEDANT will only accept surface sources at either left or right (or both) external boundaries of the physical problem.

The units on the surface source are the same as those for angular flux.

G. Input of Quadrature Sets

The ONEDANT code package has the option of obtaining the discrete-ordinates angular quadrature coefficients from a SNCONS standard interface file^{1,2}, from one of two built-in sets in subroutine SNCON, or from card-image input. The input parameter IQUAD in BLOCK V of the card-image input specifies the source of these coefficients. The number of quadrature coefficients, MM, is determined from the input S_N order parameter ISN and the geometry specification input parameter IGEOM, both found in input BLOCK I. Values of MM are shown in Table VI.

The built-in constants provided in the code are (i) the Gaussian P_N constants (IQUAD=1) for $S_2, S_4, S_6, S_8, S_{12}, S_{16}, S_{20}, S_{24}, S_{32},$ and S_{48} , (ii) the double Gaussian DP_N constants (IQUAD=2) for $S_4, S_8, S_{12}, S_{16}, S_{24}, S_{32}, S_{40}, S_{48}, S_{64},$ and S_{96} , and (iii) generalized quadrature, GQ_N , constants (IQUAD=4) for S_4, S_8, S_{12} and S_{16} . For most problems the P_N set is satisfactory. For thin-slab problems in which the angular representation for the leakage flux is important, the DP_N set is recommended. For cylindrical or two-angle plane calculations with anisotropic scattering, the GQ_N set is recommended. The GQ_N set for cylinders and two-angle planes is a generalized even-moment EQN quadrature set. The generalized quadrature capability in the code was developed and provided by James E. Morel, Sandia National Laboratories, Albuquerque, New Mexico.

For problems with anisotropic scattering, it is important that the S_N order be chosen sufficiently large such that the spherical harmonic polynomials described in Section A of this chapter are correctly integrated. Otherwise, the numerical quadrature error may introduce nonphysical contributions to the neutron balance, preventing convergence of the problem to the desired precision.

For user card-image input of S_N quadrature sets through the WGT and MU arrays in BLOCK V, it is necessary that the sets be correctly ordered as illustrated in Figs. 7, 8, and 9. In addition, if the sums $1 - \sum_m w_m$, $\sum_m \mu_m$, and $\sum_m w_m \mu_m$ exceed 10^{-5} , an error message is printed. It should be noted that if the user provides the card-image input arrays WGT and MU, the code will use these arrays for the quadrature constants irrespective of the value of IQUAD entered in the input, i.e., the WGT and MU input arrays will override any other source quadrature constants.

H. Boundary Conditions

Several boundary condition options are available to the user of ONEDANT as follows:

- Vacuum boundary condition — the angular flux on the boundary is identically zero for all incoming directions.
- Reflective boundary condition — the incoming angular flux on the boundary is set equal to the outgoing angular flux in the direction corresponding to specular reflection.
- Periodic boundary condition — the incoming angular flux on one boundary is set equal to the outgoing angular flux in the same direction on the opposite boundary.
- White boundary condition — the incoming angular fluxes on the boundary are each set equal to the single value chosen such that the net flow across the boundary is zero, i.e.,

$$\psi_{\text{incoming}}^{(m)} = \frac{\sum w_n \mu_n \psi(\mu_n)_{\text{outgoing}}}{\sum w_n \mu_n}$$

where the sums range over all outgoing directions. This condition is used primarily for cell calculations in cylindrical and spherical geometries where it is applied to the right (outer radial) boundary.

The above boundary conditions are controlled by the BLOCK V input parameters, IBL (left boundary), and IBR (right boundary). For planar geometries (IGEOM=1), both IBL and IBR must be specified. For curvilinear geometries (IGEOM=2 or 3), only IBR need be specified since the left boundary is assumed by the code to be at the radial origin ($r=0$), for which the curvilinear geometry, $r=0$ boundary condition is the only physical condition possible.

Note: Use of a reflective boundary condition (IBL or IBR=1) requires the S_N quadrature set to be symmetric about $\mu = 0$.

Two additional boundary conditions, not controlled uniquely by IBL/IBR, are the albedo and surface source conditions defined as follows:

- Albedo condition — the incoming angular flux on the boundary is set equal to a user-supplied albedo times the value it would have without the albedo. It may be used in conjunction with either the reflective or white boundary condition described above. The use of albedoes is controlled solely by the presence of the LBEDO and/or RBEDO array specifications in the BLOCK V and card-image input.
- Surface source boundary conditions — the incoming angular fluxes on the boundary are set equal to the user-supplied values as specified in the SILEFT or SALEFT and/or SIRITE or SARITE input arrays in BLOCK V. See discussion of inhomogeneous sources in Sec. F of this chapter.

I. Buckling Corrections

Leakage from the tranverse dimension(s) of a multidimensional system may be simulated by using a user-specified buckling height (BHGT) and/or buckling width (BPTH) in the BLOCK V card-image input. For plane and 2-angle plane geometries (IGEOM=1), both BHGT and BPTH may be specified. For cylindrical geometry only the buckling height, BHGT, may be specified. The buckling dimensions are in units consistent with the units on cross sections, e.g., in cm if cross sections are in cm^{-1} . If diffusion theory is assumed adequate, then the flux shape in the transverse direction, say Z, is of the form

$\cos \pi Z/\tilde{h}$ so that the flux shape function vanishes at the extrapolated system half-heights $\pm \tilde{h}/2$. Applying this to the transport equation the transverse leakage appears as a buckling absorption with a buckling absorption cross section

$$\sigma_{a, \text{BHGT}} = \frac{\sigma}{3} \left[\frac{\pi}{\sigma^* \text{BHGT} + 1.4209} \right]^2,$$

where σ is the macroscopic zone total cross section, BHGT (or similarly BWTH) is the buckling height (or buckling width), and $1.4209/\sigma$ is twice the Milne planar extrapolation distance.

The buckling absorption correction is applied to both the total cross section and absorption cross section for each group and zone in the physical problem. Consequently, the absorption rate printed in the output SOLVER module coarse-mesh balance table contains this buckling absorption.

1. Eigenvalue Searches

It is possible in ONEDANT to perform an eigenvalue search on material concentration (concentration search), system dimensions (dimension search), or the time absorption (alpha search) to achieve a desired value of k_{eff} . The type of search is controlled by the input parameter IEVT supplied in BLOCK V of the card-image input as follows:

<u>IEVT*</u>	<u>Type of Eigenvalue Search</u>
2	Time absorption (alpha)
3	Concentration
4	Critical size (dimension)

*Not included here are the options IEVT= -1 for inhomogeneous source problems in a fissionable system, IEVT = 0 for inhomogeneous source problems in nonfissionable systems, and IEVT = 1 for k_{eff} calculations.

For time-absorption calculations, the time-dependent angular flux is assumed to be separable in time and space, viz.,

$$\psi(r, \underline{\Omega}, t) = \psi(r, \underline{\Omega}) e^{\alpha t} .$$

If this assumption is inserted into the time-dependent transport equation, the exponentials cancel and a fictitious cross-section term of the form α/v_g appears as a correction to the total and absorption cross sections. Here v_g is the neutron speed associated with energy group g . The exponential factor α is then the eigenvalue sought in the time-absorption eigenvalue search. Obviously, $\alpha = 0$ for an exactly critical system, and $\alpha > 0$ for a super-critical system.

For concentration searches, the Material concentrations are modified in accordance with the description provided under the ASGMOD array in BLOCK IV of the card-image input (see Ch. IV. C).

For dimension searches, the coarse-mesh boundaries can be modified selectively to obtain a critical system. The modified coarse-mesh boundaries, \tilde{R}_k , are calculated from the initial input boundaries, R_k , by

$$\tilde{R}_{k+1} = \tilde{R}_k + (R_{k+1} - R_k) * (1 + EV * RM_k), \quad (36)$$

$$k = 1, 2, \dots, IM,$$

where EV is the eigenvalue sought in the search. The factors RM_k are the coarse-mesh radii modifiers which are input by the user via the RM array in the BLOCK V card-image input, and control how the coarse-mesh boundaries are modified. Clearly, if RM_k is zero, the thickness of the k^{th} zone is not altered. If all RM_k are unity, the system dimensions are uniformly expanded ($EV > 0$) or contracted ($EV < 0$). Many sophisticated changes can be made, limited only by the ingenuity of the user. For example, an interface between two zones may be moved while the remainder of the system is left unchanged.

In all three types of searches the appropriate system parameter may be adjusted to achieve the desired value of k_{eff} . This value is taken to be unity

(criticality) unless the input parametric value type (IPVT in BLOCK V of the card-image input) is set to unity. If IPVT = 1, the desired parametric value of k_{eff} is input by the user as PV (in BLOCK V).

For concentration searches (IEVT = 3) and dimension searches (IEVT = 4), it is also possible to adjust the appropriate system parameter to achieve a system whose neutral particle flux is changing exponentially in time at the rate $e^{\alpha t}$ by setting the input parametric value type, IPVT, to 2. In this case the user enters the desired exponential factor α as the parametric value PV in the input. Note that an α of 0.0 corresponds to a normal concentration or dimension search on a k_{eff} of unity.

It is important to recognize that the value of PV input by the user remains fixed throughout the search process.

Regardless of the parameter being adjusted, the search is executed by performing a sequence of k_{eff} -type calculations, each sequence for a different value of the parameter being treated as the eigenvalue. The search is for a value of the parameter that makes the value of λ unity where λ is defined as

$$\lambda = \frac{(\text{Fission source})^k + \text{Inhomogeneous source}}{(\text{Fission source})^{k-1} + \text{Inhomogeneous source}} \quad (37)$$

for the k^{th} outer iteration. The search is controlled by the subroutine NEWPAR in the SOLVER module.

In the following description of NEWPAR, it is helpful to refer to Fig. 10, in which the deviation of λ from unity for each outer iteration is plotted.

For the initial system, NEWPAR continues the outer iteration until two successive values of λ differ by less than EPS0. For subsequent sequences of λ values, a different convergence precision, XLAX, is used. After the first converged λ sequence is obtained, the initial value of the eigenvalue (EV) is altered by EVM, an input value. If $\lambda > 1$ (multiplying system), the new eigenvalue is equal to $EV + EVM$; if $\lambda < 1$ (decaying system), the new value is $EV - EVM$. Thus, the sign and value of EVM should be chosen such that the use of $EV + EVM$ will reduce the reactivity of the system. Conversely, the use of $EV - EVM$ should increase the reactivity of the system.

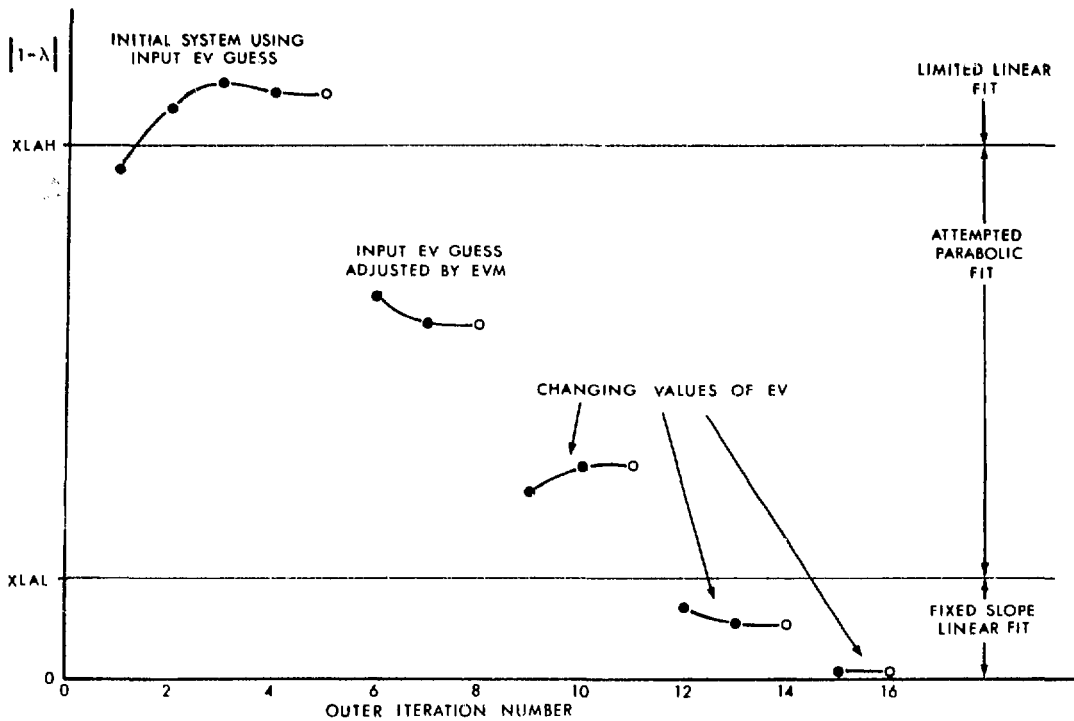


Fig. 10. Variation of λ during a hypothetical eigenvalue search.

Basically, after two converged values of λ are obtained for two different system configurations, subroutine NEWPAR attempts to fit a curve through the most recent values to extrapolate or interpolate to a value of unity. Depending on the amount of information available and the size of $|1 - \lambda|$, this fit proceeds in different ways. A parabolic fit cannot be made until three converged values of λ are available, and is not attempted unless $|1 - \lambda|$ is greater than an input search lower limit (XLAL) and less than an input search upper limit (XLAH). If a parabolic fit is tried and the roots are imaginary, a straight-line fit is used. If the roots are not imaginary, the closest root is used as the new value of EV. Once a bracket is obtained (change of sign of $\lambda - 1$), the fit procedure is not allowed to move outside the region of the bracket. Should

a parabolic fit select an eigenvalue outside the bracket region, this value is rejected and the new value is taken to be one-half the sum of the previous value and the value previous to that.

Whenever the parabolic fit is not used (i.e., $|1 - \lambda| < XLAL$), a linear fit is used and the new eigenvalue is computed from

$$(EV)_{\text{new}} = (EV)_{\text{old}} + \text{POD} * \text{EVS} * (1 - \lambda) \quad , \quad (38)$$

where POD is an input "parameter oscillation damper" that may be used to restrict the amount of change in the eigenvalue. In Eq. (38), EVS is a measure of the slope of the curve. When $|1 - \lambda| > XLAH$, $(1 - \lambda)$ in Eq. (38) is replaced by XLAH (with the correct sign) to prevent too large a change in EV. After $|1 - \lambda| < XLAL$, the value of EVS is fixed and kept constant until convergence to prevent numerical difficulty in the approximation of the derivative when λ is close to unity.

Because parametric search problems represent sequences of k_{eff} calculations, it behooves the user to study the use of subroutine NEWPAR in order to optimize his calculations. It also behooves the user to pose soluble problems. That is, there are many problems, especially concentration searches, for which solutions are not possible, and discovering this by trial and error is the hard way. Ideally, the user will have some estimate of the critical parameter available from a lower order computation.

Convergence in time-absorption calculations is typically one-sided. If EV (the eigenvalue α) is negative, then there is a possibility that the corrected total cross section will become negative. If this happens, the automatic search procedure may fail dramatically. For this reason $\text{POD} = 0.5$ or less is frequently used in such searches.

K. Adjoint Computations

The ONEDANT code package solves the adjoint transport equation by transposing (in energy) the matrices of scattering cross sections and inverting the group order of the problem. The transposition of the scattering matrix converts a downscatter problem to an upscattering problem so that by inverting the group order the problem will execute in a downscatter-like mode. In addition to transposing the scattering matrices, the fission source term in the transport

equation is transposed so that instead of $\chi_g \sum_h (v\sigma_f)_h \phi_h$, one has $(v\sigma_f)_g \sum_h \chi_h \phi_h$. The code does not transpose the angular direction matrix associated with the leakage terms in the transport equation. Instead, the adjoint calculation of the leakage operator proceeds as in the direct (forward) calculation, but the results of the adjoint calculation for direction $\underline{\Omega}$ must be identified as the adjoint solution for direction $-\underline{\Omega}$. For example, the vacuum boundary condition at a surface (no incoming angular flux) in an adjoint calculation must be interpreted as a condition of no outgoing flux. Likewise, the adjoint leakage at a surface must be interpreted as incoming instead of outgoing.

All group-order inversions and fission source and scattering matrix transpositions are performed by the code; the user need only set the input parameter ITH in BLOCK V to unity to effect an adjoint calculation. (If the problem contains inhomogeneous sources, these sources must quantitatively be, of course, the adjoint sources.)

The printed output from the SOLVER module in an adjoint calculation indicates the correct group ordering and need not be inverted by the user. The adjoint fluxes from the SOLVER module are written to a binary ATFLUX standard interface file. The ATFLUX file description is shown in Ref. 1.

In performing the adjoint reversals of the scattering matrices and the group inversions, the code prepares a binary, code-dependent interface file ADJMAC. This ADJMAC file contains the adjoint-reversed Material cross sections to be used by the SOLVER module. ADJMAC is essentially the adjoint-reversed counterpart to the MACRXS file described in Sec. D.1 of this Chapter, and the rules for saving and using an existing ADJMAC file are the same as for an existing MACRXS file.

The performance of adjoint edits is described in Ch. VII.

L. The MAXLCM Parameter

Through the use of the input parameter, MAXLCM, in BLOCK I of the card-image input, the Los Alamos National Laboratory user can specify the maximum amount of large core memory (LCM) he wishes to use. If unspecified, the value of MAXLCM is defaulted to 50 000₁₀ words.

The modular structure of ONEDANT is such that the processing of each input BLOCK, as well as the SOLVER and EDIT modules, each uses LCM storage independently and each such stage requires a different amount of LCM. (In most cases

the cross-section processing stage and the SOLVER module require the greatest amount of LCM.) At each stage the amount of LCM required for that stage is computed with random disk usage, if necessary, such that the stage requires no more than MAXLCM words of LCM. The computing system is instructed to reserve only the actual amount of LCM needed to perform each stage. Thus, if a value of MAXLCM = 100 000 has been input, but a particular stage requires only 30 000 words of LCM, the computing system will reserve only the 30 000 words until that stage is completed. The code will then re-specify the 100 000 allocation and proceed to the next stage. In the Los Alamos time-sharing computing environment, this ability to expand and contract the LCM requirements to the amount actually needed by each stage can be quite advantageous.

The user must be cautioned against specifying too small a value of MAXLCM since the result may be an excessive use of random disk, the access to which is relatively time-consuming.

VI. DETAILS RELATED TO SOLVER MODULE EXECUTION

This chapter provides details pertinent to the execution of the SOLVER module of ONEDANT in which the time-independent, multigroup form of the one-dimensional, diffusion synthetic accelerated, discrete-ordinates transport equation is solved. Described are the iteration strategy, convergence criteria, termination criteria for the iterative loops, and iteration monitor print.

A. Iteration Strategy

As described in Ch. II.D of this report, the ONEDANT SOLVER module employs the diffusion synthetic method to accelerate the iterative procedure used in solving the transport equation. In this section is described the iteration strategy used in the execution of the SOLVER module and reflected in the iteration monitor printout supplied as printed output.

The basic features of the iteration strategy are shown in the simplified flow diagram of Fig. 11. As indicated, there are two different iterative procedures, one for problems containing fissionable material and/or energy-group up-scattering and one for problems with neither fissions nor upscattering.

The iterative strategy is divided into two parts: inner iterations and outer iterations. The inner iterations are concerned with the convergence of the pointwise scalar fluxes in each group for a given source distribution. The outer iterations are concerned with the convergence of the eigenvalue, the fission source distribution and the energy-group upscatter source if any or all are present.

For problems containing fissionable material the iterative procedure begins with the calculation of a diffusion coefficient for each space-energy point using

$$D(x,g) = \begin{cases} 1/[3\Sigma_t(x,g)] & , \quad \text{isotropic scatter} \\ 1/\{3[\Sigma_t(x,g) - \Sigma_{s1}(x,g \rightarrow g)]\} & , \quad \text{anisotropic scatter} \end{cases} \quad (38)$$

where $D(x,g)$ is the diffusion coefficient at position x for energy group g , $\Sigma_t(x,g)$ is the macroscopic total cross section at the space-energy point in question, and $\Sigma_{s1}(x,g \rightarrow g)$ is the P_1 anisotropic self-scatter cross section. It should be noted that $\Sigma_t(x,g)$ is formed from the isotope cross

sections contained in the total cross-section position in the cross-section library. The data provided in this position may, in fact, contain the transport cross section in transport-corrected cross-section libraries for isotropic scatter.

Using the above diffusion coefficients, a standard diffusion calculation is performed. Beginning with a flat fission source guess, the diffusion fluxes are calculated for each energy group. With the fluxes for all groups, a new fission source rate distribution, $F(x)$, is calculated. This new $F(x)$ is then used to generate new diffusion fluxes. The process is repeated until both $F(x)$ and the pointwise fluxes are converged. Each such recalculation of $F(x)$ is called a DIFFUSION SUB-OUTER ITERATION.

Next, using the diffusion-converged $F(x)$ and using the first energy-group diffusion scalar fluxes to fix the within-group scattering sources, a single discrete-ordinates transport sweep through the spatial mesh is made for the first energy group. In this sweep angular fluxes are generated. Using Fick's Law, Eq. (6), these angular fluxes are used to calculate an effective diffusion coefficient, $\underline{D}(x)$, at each mesh point. With these effective diffusion coefficients, a diffusion sweep for the group is performed to determine the group scalar flux, $\phi(x)$, at each point. This transport sweep, followed by a calculation of $\underline{D}(x)$, followed by a diffusion sweep is called an INNER ITERATION. Since the new diffusion calculated group scalar flux, $\phi(x)$, changes the within-group scattering source term, the user may opt to perform another (or several more) inner iterations before proceeding to the next energy group. The IITL and IITM input parameters in BLOCK V of the input control the maximum number of inner iterations to be performed for each group. Normally, however, the user need not enter these parameters since the code will use default values that have been found to be reliable.

When the inner iterations for the first energy group are completed, the group scalar fluxes and flux moments are used to calculate the scattering source for the next group. One or more inner iterations are performed for the next group and the process is repeated until all energy groups have been completed.

When all energy groups have been calculated via inner iterations, the group fluxes are used to calculate a new fission source rate distribution, $F^0(x)$. Following this a series of diffusion sub-outer iterations is performed. In

these diffusion sub-outer iterations, however, the effective diffusion coefficients from the last-completed inner iteration for each group are used, thus making the sub-outer iteration calculation a synthetic diffusion calculation.

Each completion of the diffusion sub-outer iteration process based on the current set of diffusion coefficients defines the completion of an OUTER ITERATION (see Fig. 11). By the initial definition of $D(x)$, Eq. (38), the first outer iteration is seen to be a pure diffusion calculation, while all subsequent outer iterations are synthetic diffusion in nature. Outer iterations continue until convergence (as described in Sec. B of this chapter) is achieved. The above procedure, of course, employs the diffusion coefficient correction scheme described in II.D.

For problems containing an inhomogeneous source (either distributed in space or localized as a surface source, or both), the iteration solution strategy may take one of two forms. The form used is controlled by the input parameter IEVT in BLOCK V of the input. When IEVT= -1 the iterative procedure used is the same as described above. IEVT= -1 should be used if the inhomogeneous source problem contains either fissionable material or material capable of producing energy-group upscattering of particles. If the inhomogeneous source problem contains neither fission nor upscattering, a value of IEVT= 0 should be selected. In this case the source correction scheme is used in the diffusion synthetic acceleration and the iterative strategy is as shown in Fig. 11. No outer iteration is performed and problem convergence is determined by the inner iteration.

B. Convergence Criteria

The convergence of the iterations is monitored at both the inner and the outer iteration level. The input parameters that control the number of iterations are EPSI, EPSO, IITL, IITM, and OITM found in BLOCK V of the SOLVER module input.

1. Inner Iteration Convergence. The inner iterations for a given energy group are said to be converged when the pointwise scalar fluxes from one inner iteration to the next satisfy the condition:

$$\max \left| \left(\phi_{i,g}^{\ell} - \phi_{i,g}^{\ell-1} \right) / \phi_{i,g}^{\ell} \right| < \text{EPSI} \quad (39)$$

where $\phi_{i,g}^{\ell}$ is the scalar flux for mesh point i , group g , and inner iteration ℓ , and where EPSI is the user-input inner iteration convergence criterion.

2. Diffusion Sub-Outer Iteration Convergence. The convergence of the diffusion sub-outer iterations requires the satisfaction of two criteria. Let us use the index v to denote the outer iteration number and the index p to denote the diffusion sub-outer iteration number. Convergence of the diffusion sub-outers is then satisfied when both

$$\max \left| (\phi_{i,g}^{p,v} - \phi_{i,g}^{p-1,v}) / \phi_{i,g}^{p,v} \right| < 0.95 * \text{EPSX} \quad (40)$$

$$\text{and} \quad \left| 1 - \lambda_d^{p,v} \right| < \text{EPSO} \quad (41)$$

In the above

$$\text{EPSX} \equiv \text{EPSI} * [1 + \text{NGROUP} * \exp(-100 * \text{EPSI})] \quad , \quad (42)$$

where NGROUP is the number of energy groups, and

$$\lambda_D^{p,v} \equiv (F^{p,v}, 1) / (F^{p-1,v}, 1) \quad . \quad (43)$$

The notation (F,G) denotes the inner product, or volume integral, of the product $F * G$.

3. Full Convergence. Full, or overall problem, convergence is achieved for problems requiring outer iterations when Eqs. (40) and (41) are satisfied together with the additional requirements that

$$\max \left| (\phi_{i,g}^{1,v} - \phi_{i,g}^{p^*,v-1}) / \phi_{i,g}^{1,v} \right| < \text{EPSX} \quad (44)$$

$$\text{and } |1 - \lambda^v| < \text{EPSO} \quad (45)$$

where $\phi_{i,g}^{1,v} \equiv$ scalar flux at point i , group g from the first diffusion sub-outer iteration for outer iteration v .

$\phi_{i,g}^{p*,v} \equiv$ scalar flux at point i , group g from the last diffusion sub-outer iteration of outer iteration $v-1$

and

$$\lambda^v \equiv (F^{0,v},1)/(F^{0,v-1},1) \quad (46)$$

In Eq. (46), $(F^{0,v},1)$ denotes the volume integral of the fission source rate distribution calculated at the end of the inner iteration cycle but before the performance of diffusion sub-outer iterations for outer iteration v (see Fig. 11).

Note that the pointwise flux convergence condition of Eq. (44) is a measure of the pointwise flux change from one outer iteration to the next -- a rather stringent criterion.

4. Iterative Loop Termination. Each of the iterative loops (inner iterations, diffusion sub-outer iterations, and outer iterations) is terminated when either the convergence criteria for that loop are met or when a specified maximum number of iterations have been attained.

For inner iterations the number of iterations is limited by the user input parameter IITL. If the user elects to omit this quantity, the code chooses an appropriate default value.

In problems where outer iterations are not required, that is, fixed-source problems with IEVF = 0, the value of IITL is usually chosen to be large, say 20-50, in order that the pointwise fluxes be allowed to meet the convergence criterion before the number of inner iterations reaches IITL.

For eigenvalue problems (IEVF > 0) the usual procedure is to allow only one inner iteration per group until the fissions, upscatter sources, and diffusion scalar fluxes have neared full convergence. When this is achieved, the allowable number of inner iterations is increased to IITM (a user input quantity) which typically is in the range of 10-20 in order to permit full convergence of the transport fluxes. The assumption here is that it is most efficient to

first converge the fission/upscatter sources and then to converge the pointwise fluxes. The code will switch the inner iteration limit from IITL to IITM when both

$$\left| 1 - \lambda_D^{p,v} \right| < 3*EPSO \quad (47)$$

and

$$\max \left| (\phi_{i,g}^{p,v} - \phi_{i,g}^{p-1,v}) / \phi_{i,g}^{p,v} \right| < 10*EPSX \quad (48)$$

where $\lambda_D^{p,v}$ is defined by Eq. (43) for diffusion sub-outer iteration p , outer iteration v and $EPSX$ is given by Eq. (42).

For fixed-source problems with fission and/or upscatter ($LEV T = -1$), a procedure similar to that for eigenvalue problems is followed except that IITL is usually chosen to be in the range of 2-5 instead of 1.

Diffusion sub-outer iterations are terminated when either the convergence criteria are met or when 100 sub-outers have been performed. The maximum allowable number of sub-outers is built into the code and is not under user control.

Outer iterations are terminated when either the full convergence criteria are met or when the number of outer iterations reaches OITM, a user-input quantity. If not supplied by the user, the code will default the value of OITM to 20.

C. Iteration Monitor Print

In the printed output from the SOLVER module, an iteration monitor print is supplied for the user. The user should always inspect this monitor print to determine whether or not the problem has successfully converged.

At the end of each outer iteration the monitor provides the elapsed computer time in seconds, the outer iteration number, and the number of diffusion sub-outer iterations required. A number of sub-outer iterations of 100 implies that the diffusion sub-outer iteration did not converge to the criterion of Eqs (40) and (41) before reaching the maximum allowable number of sub-outer iterations. Also provided is a message as to whether or not the inner iterations satisfied

their convergence criterion, Eq. (39). Finally are included the values of $\lambda^v - 1$ and the maximum pointwise flux error corresponding to the values used in the test for full convergence given by Eqs. (45) and (44), respectively.

In addition to the basic outer iteration information described above, the monitor print provides an inner iteration monitor for certain outer iterations. This inner iteration monitor is always provided for a fixed-source problem without fission or upscatter (IEVT= 0) since only one outer iteration is required. For other problems (IEVT \neq 0) the inner iteration monitor is only provided for outer iterations following the satisfaction of the "nearly converged" conditions of Eqs. (47) and (48). In the inner iteration monitor are included the group number, the number of inner iterations taken, the maximum pointwise scalar flux error (see Eq. (39)), and the spatial mesh point where this maximum error occurred.

In the inner iteration monitor two warning messages are available to the user. A message "TRANSPORT FLUXES BAD" is provided when nonpositive transport scalar fluxes exist following the last inner iteration. The presence of nonpositive scalar fluxes causes the diffusion inner iteration acceleration to be disabled. Although such a condition is not necessarily fatal, it does usually indicate that the spatial mesh is too coarse and that the results will be suspect. The second message, "ACCELERATION DISABLED", is provided when the transport correction to the diffusion coefficient, diffusion source, or diffusion removal term is such that the synthetic diffusion equation cannot be applied to accelerate that inner iteration. The presence of the message does not necessarily make the answers suspect if convergence is achieved; it merely tells the user that the inner iteration could not be diffusion accelerated.

VII. DETAILS RELATED TO EDIT MODULE INPUT AND EXECUTION

The basic function of the EDIT module is to perform postprocessing, or edit, operations using multigroup, pointwise scalar fluxes generated in a previous execution of the SOLVER module or, perhaps, in some other neutronics code. The EDIT module uses the scalar fluxes, multiplies them by suitable quantities hereafter called response functions, calculates sums of these products over space and/or energy (if desired), and produces printed output of the results. Recall, from Ch. II of this manual, that the EDIT module is essentially a free-standing code module accepting only interface files as input. Most of these interface files are general in nature in that they apply both to the SOLVER and the EDIT modules (see Fig. 1 in Ch. II). Included in these general files are the geometry specifications (GEODST file), the material mixing and cross-section specifications (NDXSRF, ZNATDN, and SNKEDT files), and the assignment of materials to zones specifications (ASGMAT file). The input requirements and details related to these specifications and files are provided in Ch. V. Another general file required by the EDIT module is a standard scalar flux interface file, either regular (forward) scalar fluxes (RTFLUX file), or adjoint scalar fluxes (ATFLUX file). Either an RTFLUX or an ATFLUX file is automatically provided by the SOLVER module when it is executed. The specific edit operations to be performed using the information from the above general files are provided to the EDIT module by means of an EDITIT interface file. This file is created by the INPUT module solely from user card-image input in BLOCK VI of the input data.

Because of the structure and interface file linkage of the ONEDANT code, several different EDIT module runs can be performed using the same set of general files. For example, once the SOLVER module is executed and its scalar flux interface file saved, the EDIT module can be repeatedly executed without re-execution of the SOLVER module. Only the EDIT module card-image input need be changed so that a new EDITIT file is created between runs.

The remainder of this chapter provides details pertinent to the editing options available to the user in the EDIT module card-image input (BLOCK VI).

A. Spatial Options for Edits

Edits can be performed on the fine spatial mesh points (as specified in BLOCK II geometry input) or on integrals over specified spatial intervals (called Edit Zones).

The fine space-point option is chosen by setting the BLOCK VI input parameter PTED to unity. In this form the edit quantity, denoted by ρ , for the i th spatial mesh point is computed as

$$\rho_{i,g'} = \sum_{g \in g'} \phi_{i,g} R_{i,g}$$

where

- $\phi_{i,g}$ = scalar flux for mesh point i , energy group g
- $R_{i,g}$ = a response function which may be either input directly via the RSFE and RSFX arrays (below) or formed from input cross sections
- g' denotes an Edit energy-broad-group (See VII. B) consisting of one or more SOLVER energy groups.

With the BLOCK VI input parameter BYVOLP set to unity, the above edit quantity will be multiplied by the mesh interval "volume" V_i . The user may also select those points, or intervals, for which he wishes point edits by use of the POINTS input array in BLOCK VI. If PTED=1 and the POINTS array is not specified, the code will provide output for all mesh points (default).

To obtain edit quantities that are integrals over desired spatial intervals, the input quantity ZNED is set to unity. The desired spatial intervals, called Edit Zones, are specified by the user through the EDZONE array in input BLOCK VI. In specifying the Edit Zones through the EDZONE array the following rules must be observed:

- (i) each and every fine-mesh interval (point) must be assigned to an Edit Zone, that is, given an Edit Zone number,
- (ii) Edit Zone boundaries are arbitrary, that is, they are independent of coarse-mesh or material boundaries,
- (iii) Edit Zone numbers must be positive integers in the range 1,2,...,N where N is the total number of Edit Zones desired.

Example: Given a problem with 30 mesh intervals. It is desired that edit quantities be produced that are integrals over the first 10 mesh intervals, the second 10 mesh intervals, and the remaining 10 mesh intervals. There are thus 3 Edit Zones each comprising 10 mesh intervals. Using the free-field repeat option of Table I, the EDZONE specification could be provided as EDZONE= 10R1, 10R2, 10R3 to specify that the first 10 space intervals are in Edit Zone 1, the second 10 in Edit Zone 2, and the third 10 in Edit Zone 3. It should be noted that the ordering of the Edit Zones 1,2,3 with the first, second, and third set of 10 mesh points is not required.

Thus, with ZNED=1, the EDIT module will produce edit quantities, ρ , for Edit Zone Z_m as

$$\rho_{Z_m, g'} = \sum_{g \in g'} \sum_{i \in Z_m} \phi_{i, g} R_{i, g} V_i .$$

If Edit Zone edits are requested (ZNED=1) and the EDZONE array is not specified, the code will assume a default specification of the Edit Zones equal to the Coarse-Mesh intervals (see XMESSH input array in BLOCK II).

IMPORTANT NOTE: In order to get printed output from the EDIT module, either point edits (PTED=1) or edit zone edits (ZNED=1) or both must be specified.

B. Energy-Group Options for Edits

The user may select the energy-group structure desired for the edit output by means of the ICOLL input array in BLOCK VI. Through this input array the user can collapse the energy-group structure used in the SOLVER module down to fewer (broader) groups for edit purposes.

Example: Consider a 24 energy-group structure used by the SOLVER module in which the first 12 groups are considered "fast" groups, groups 13 through 21 are "epithermal" groups, and groups 22

through 24 are "thermal" groups. If it is desired that edit quantities be calculated as integrals (sums) over the three broad groups denoted fast, epithermal, and thermal, the ICOLL array would be specified as ICOLL=12, 9, 3 to collapse the first 12 groups into Edit Energy-Broad-Group 1 (the "fast" broad group), the next 9 groups (groups 13 through 21) into Edit Energy-Broad-Group 2 (epithermal), and the last 3 groups (groups 22 through 24) into Edit Energy-Broad-Group 3 (thermal).

If the ICOLL array is not specified, the code will assume the default condition of one SOLVER module energy group per Edit Energy-Broad-Group.

The IGRPED input parameter in BLOCK VI is used to control the printed output with respect to the Edit Energy-Broad-Groups. With IGRPED=0 only the energy-group total (sum over all groups) of the edit quantities is printed. With IGRPED=1 or =2 edit quantities for each of the Edit Energy-Broad-Groups are printed. With IGRPED=3 edit quantities for each Edit Broad Group plus the energy-group total are printed.

C. Forms of Response Functions

As indicated in the preceding sections, edit quantities all involve taking the product of the scalar flux, $\phi_{i,g}$, and a response function, $R_{i,g}$, (for spatial mesh point i and energy group g). In this section are described the various forms that the response function $R_{i,g}$ can take.

1. Cross-Section Response Functions: EDXS Input Array. Response functions can be formed directly from cross-section data. In this case it is necessary to specify the particular type, or types, of cross sections to be used, that is, (n,γ) , (n,α) , total, absorption, etc. The cross-section data provided to the EDIT module on the SNXEDT file will contain a particular cross-section type in a unique position within the cross-section data table as indicated in Table II. Through the EDXS input array in the BLOCK VI input, the user specifies which cross-section types are desired using either the integer edit position numbers or the Hollerith names as given in Table II.

Example: Consider a problem in which Isotope cross sections were supplied by means of an ISOTXS binary file. It is desired that edits be performed using both the n,α and n,γ cross sections. Using Table II, the EDXS array would be input as EDXS= N-ALPH, N-GAMM or, alternatively, as EDXS= 8, 10.

The specific forms of cross-section-based response functions available in the EDIT module are the resident macroscopic, Isotope microscopic, constituent, and Material forms. Each of these is described below.

a. Resident Macroscopic Cross Section Response Functions: RESDNT Input Parameter. The resident macroscopic cross section, $\Sigma_{i,g}^{RES}$, at mesh point i , energy-group g is defined as the actual macroscopic cross section that was used by the SOLVER module. To obtain this response function, namely

$$R_{i,g} = \Sigma_{i,g}^{RES} ,$$

the BLOCK VI input parameter RESDNT is set to unity.

b. Isotope Microscopic Cross-Section Response Functions: EDISOS Input Array. Isotope microscopic cross section, σ_g^{ISO} , may be used for the response functions by identifying the isotopes desired through the EDISOS BLOCK VI input array. In this edit the cross sections are taken directly from the EDIT module file SNXEDT, which themselves originally came from the basic cross-section library (ISOTXS or GRUPXS file or BCD card-image library). Note that the response function

$$R_{i,g} = \sigma_g^{ISO}$$

is spatially constant so that the edit quantity $\sigma_g^{ISO} \phi_{i,g}$ will be calculated at mesh point i even if the isotope was not physically present at that location.

c. Resident Constituent Cross-Section Response Functions: EDCONS Input Array. Resident constituent cross sections, $\Sigma_{i,g}^{ISO}$, can be used for response functions by identifying the Isotopes desired through the EDCONS BLOCK VI input array. The resident constituent or simply, constituent, cross section is a partial macroscopic cross section given by the product of the Isotope microscopic cross section times the actual atom density associated with that Isotope at the spatial location as seen by the SOLVER module. Thus, for a constituent cross-section edit the response function $R_{i,g}$ is

$$R_{i,g} = N_i^{ISO} \sigma_g^{ISO} = \Sigma_{i,g}^{ISO}$$

for spatial mesh interval i , group g .

d. Material Cross-Section Response Functions: EDMATS Input Array. Material macroscopic cross sections, Σ_g^{MATL} , can be used for the response function by identifying the desired materials through the EDMATS array in the BLOCK VI input. In this Material edit the macroscopic cross sections for the Materials specified in the MATLS array in the Mixing input block (BLOCK IV) are reformed using the microscopic cross sections on file SNXEDT together with the mixing instructions stored on the NDXSRF and ZNATDN standard interface files. Thus, for a Material cross-section edit the response functions are of the form

$$R_{i,g} = \Sigma_g^{MATL} .$$

Note that these response functions are spatially constant so that the edit quantity $\Sigma_g^{MATL} \phi_{i,g}$ will be calculated at each mesh point, i , even if the Material was not physically present at that location.

2. User-Input Response Functions: The RSFE and RSFX Input Arrays. In addition to response functions based on cross-section data, the user may directly input response functions in a space-energy separable form through the BLOCK VI

input arrays $RSFE_g$ and $RSFX_i$ for energy-group g and space-point (interval) i . Thus, for user-input response functions,

$$R_{i,g} = RSFE_g * RSFX_i .$$

NOTE: The $RSFE$ array is required if user-input response functions are desired. The $RSFX$ input array is optional. The $RSFE$ input array can be used, for example, to obtain groupwise fluxes (or sums of groupwise fluxes) by using $RSFE$ array entries of 1.0 in the groups of interest. Fluxes can similarly be renormalized by use of the appropriate normalization factor in either the $RSFE$ or $RSFX$ arrays.

D. Response Function Summing Options

Certain response summing operations are available to the user by means of the BLOCK VI input arrays MICSUM and IRSUMS. The MICSUM array provides for the specification of cross-section response function summing, while the IRSUMS array provides for the specification of user-input response function summing. Each of these is described below.

1. Cross-Section Response Functions Sums: MICSUM Input Array. Through the use of MICSUM input array in BLOCK VI, either Isotope microscopic edit sums or resident constituent edit sums, but not both, will be computed. (Recall that Isotope microscopic edits are invoked by means of the EDISOS input array and resident constituent edits are invoked by means of the EDCONS input array.)

The MICSUM input array is a packed array with data entered as follows: a set of Isotope numbers or Hollerith names (from the basic Isotope input library) is given followed by a set of cross-section type position numbers or Hollerith names (see Table II). These sets are delimited with an entry of 0 (zero). Reaction rates (edit quantities) are calculated for each Isotope specified in the set for each cross-section type specified and summed to form the first sum. The next two sets of data are used to define the second sum, etc.

The MICSUM array is only used in conjunction with either the EDCONS array or the EDISOS array as follows:

- If the EDCONS array is specified, the summing defined by the MICSUM array applies to the resident constituent (partial macroscopic) cross sections. Isotopes used in the MICSUM array must have been used in the EDCONS array.
- If the EDCONS array is not specified and the EDISOS array is specified, the summing defined by the MICSUM array applies to the Isotope microscopic cross sections. Isotopes used in the MICSUM array must have been used in the EDISOS array.

Example: Suppose the EDCONS array were specified as

EDCONS= PU-239, PU-240, PU-241, U-238

and the MICSUM array were specified as

MICSUM= PU-239, PU-241, 0, N-GAMM, N-FISS, 0, PU-240,
U-238, 0, ABS.

For mesh point i , energy group g , the two sums specified in the MICSUM array would be

$$\text{SUM 1: } \left\{ \left[N_i (\sigma^{n,\gamma} + \sigma^f)_g \right]^{\text{PU-239}} + \left[N_i (\sigma^{n,\gamma} + \sigma^f)_g \right]^{\text{PU-241}} \right\} * \phi_{i,g}$$

$$\text{SUM 2: } \left[(N_i \sigma_g^a)^{\text{PU-240}} + (N_i \sigma_g^a)^{\text{U-238}} \right] * \phi_{i,g}$$

2. User-Input Response Function Sums: IRSUMS Input Array. Through the use of the IRSUMS input array in BLOCK VI, user-input response function edit sums can be computed. The input to the IRSUMS array is supplied as follows: a set of user-input response function numbers or names is entered and the set is delimited with an entry of 0 (zero). Edit quantities are calculated for each response function specified and the edit quantities summed to form the first sum. The next set of data is used to form the second sum, etc. Only user-input response functions that have been provided through the RSFE input array (and, optionally, the RSFX input array) can be used in the IRSUMS array.

VIII. CONTROLLING THE EXECUTION OF MODULES AND SUBMODULES

As described in Ch. II, the ONEDANT code package is comprised of three major functionally independent modules: the INPUT module, the SOLVER module, and the EDIT module. The modules are linked solely by means of binary interface files. The INPUT module processes any and all card-image input and, if required, generates the binary interface files for use by the SOLVER and/or EDIT modules. The INPUT module itself is constructed in a modular form and thus is comprised of submodules, each of which performs a unique function related to the generation of certain binary interface files. The SOLVER module accepts the appropriate interface files produced by the INPUT module (or any other computer code capable of producing such interface files), performs the transport calculation, and generates standard interface flux files for use by the EDIT module (or other computer codes). The EDIT module accepts the appropriate standard and code-dependent interface files and performs cross-section and user-input response function edits.

With the modular construction of the code package and the interface file linkage between modules and submodules, there is a great deal of flexibility provided in the execution flow of a particular computer run. For example, the processing of the input, the execution of the transport solution, and the editing of the results of the solution can be effected as three separate and distinct computer runs and not as a single (perhaps expensive) run. All that need be done is to save the appropriate interface files from each partial execution run and to make these files available to the module to be executed in the next partial execution. This mode of operation enables the user, for example, to process his problem input specification (mixing of nuclides, cross-section preparation, geometry specification, etc.,) and to analyze his input before committing it to the SOLVER module. If errors are discovered in, say, the geometry specification, the user can correct the errors in the card-image input and simply rerun the geometry-related submodule of the INPUT module. When certain that the input is correct, the user can then execute the SOLVER module. Following the successful running of the SOLVER module, one or more executions of the EDIT module can then be independently made.

In this chapter are provided details for controlling the execution of selected modules and submodules in the ONEDANT code package.

A. Module Execution Control

The execution of each of the three major modules in the ONEDANT code package (INPUT, SOLVER, and EDIT modules) can be independently controlled as described below.

1. INPUT Module Execution Control. The INPUT Module may be thought of as an interface file generating module. It processes card-image input and creates binary interface files as shown in Fig. 1 of Ch. II. Accordingly, if any BLOCK II through BLOCK VI (see Ch. IV) card-image input is provided and the BLOCK I input execution suppression flag NOFGEN is zero, the INPUT module will be executed and the appropriate interface files created.

The execution of the INPUT module will be suppressed if either of the following conditions is met: (i) the BLOCK I input parameter NOFGEN is set to unity, or (ii) there is no card-image input provided other than BLOCK I input. If the INPUT module is not executed, none of its interface files will be created in that execution of ONEDANT.

2. SOLVER Module Execution Control. Execution of the SOLVER module will be attempted if both the following conditions are met: (i) a SOLINP binary interface file exists and is available to the SOLVER module, and (ii) the BLOCK I input parameter NOSOLV is zero.

The SOLVER module will not be executed if the BLOCK I input parameter NOSOLV is set to unity.

Alternatively, since the INPUT module creates the SOLINP interface file solely from card-image input provided in BLOCK V of the input, the user can suppress the execution of the SOLVER module by simply omitting all BLOCK V data from the card-image input. On the Los Alamos and Livermore computers this procedure will suppress the execution of the SOLVER module only if there is no previously created SOLINP file in existence and available to the ONEDANT code package during its execution.

3. EDIT Module Execution Control. Execution of the EDIT module will be attempted if both of the following conditions are met: (i) an EDITIT binary interface file exists and is available to the EDIT module and (ii) the BLOCK I input parameter NOEDIT is zero.

The EDIT module will not be executed if the BLOCK I input parameter NOEDIT is set to unity.

Alternatively, since the INPUT module creates the EDITIT interface file solely from card-image input provided in BLOCK VI of the input, the user can suppress the execution of the EDIT module by simply omitting all BLOCK VI data from the card-image input. On the Los Alamos and Livermore computers this procedure will suppress the execution of the EDIT module only if there is no previously created EDITIT file in existence and available to the ONEDANT code package during its execution.

B. INPUT Submodule Execution Control (File Generation Suppression)

The INPUT module of ONEDANT is constructed in submodular form. Each submodule has a unique interface file-creation function and each has its associated card-image input. Also associated with each submodule is a BLOCK I input flag to turn off, or suppress, the execution of that submodule. The control of the execution of the INPUT module submodules is described below.

1. Geometry Submodule Execution Control. The geometry submodule creates a GEODST standard interface file¹ from BLOCK II card-image input data as described in Ch. V. B. This submodule will be executed and a GEODST file created by (i) setting (or defaulting) the BLOCK I input parameter NOGEOD to zero and (ii) providing BLOCK II input data in the card-image input "deck" or file.

The geometry submodule will not be executed (no GEODST file will be created) if (i) the BLOCK I input parameter NOGEOD is set to unity or (ii) all BLOCK II input is omitted from the card-image input "deck."

The geometry submodule consists of secondary overlay (1,2) in Fig. 2 of Ch. II.

2. Mixing Submodule Execution Control. The mixing submodule creates the standard interface files NDXSRF and ZNATDN¹ from the BLOCK IV card-image input data found in the MATLS array and, optionally, the PREMIX array as described in Ch. IV. C and Ch.V.D. This mixing submodule is contained in the secondary overlay (1,4) of ONEDANT as shown in Fig. 2 of Ch. II.

The mixing submodule will be executed and the NDXSRF and ZNATDN files created by both (i) setting (or defaulting) the BLOCK I input parameter NOMIX to zero and (ii) providing card-image input through the MATLS array in BLOCK IV.

The mixing submodule will not be executed if NOMIX is set to unity or (ii) the MATLS input array is omitted from the BLOCK IV card-image input or (iii) LIB= MACRXS in BLOCK III.

It is important to note that the mixing submodule and the NDXSRF and ZNATDN files are closely linked to the assignment-of-materials-to-zones submodule (ASGMAT interface file) and the working-cross-section-file submodule (MACRXS and SNXEDT files) described below.

3. Assignment-of-Materials-to-Zones Submodule Execution Control. The assignment-of-materials-to-zones submodule, contained in secondary overlay (1,4) of ONEDANT (see Fig. 2), creates the code-dependent interface file ASGMAT from the BLOCK IV card-image data found in the ASSIGN array. Details on the assignment of materials to zones are given in Ch. IV. C and V.E.

This submodule will be executed and the ASGMAT file created by both (i) setting (or defaulting) the BLOCK I input parameter NOASG to zero and (ii) providing card-image input through the ASSIGN array in BLOCK IV.

The submodule will not be executed (no ASGMAT file created) if either (i) the BLOCK I input parameter NOASG is set to unity or (ii) the ASSIGN input array is omitted from the BLOCK IV card-image input.

4. Working-Cross-Section-File Submodule Execution Control. The working-cross-section-file submodule creates the code-dependent interface files MACRXS and SNXEDT (described in Ch. V.D.). This submodule is contained in secondary overlays (1,3) and (1,7) or (1,5) or (1,6) depending on the form of the basic input cross-section library as shown in Fig. 2.

The working-cross-section-file submodule will be executed and the MACRXS and SNXEDT files created if both the following conditions are met: (i) the BLOCK I input parameter NOMACR is set (or defaulted) to zero, and (ii) the BLOCK III input parameter LIB is not specified as LIB= MACRXS.

The submodule will not be executed (no MACRXS and SNXEDT files created) if either (i) the BLOCK I input parameter NOMACR is set to unity or (ii) the BLOCK III input parameter LIB is specified as LIB= MACRXS.

Since the formation of the working cross-section files MACRXS and SNXEDT can be quite time-consuming for large multigroup cross-section libraries, it is frequently advantageous to save the MACRXS and SNXEDT files created in

one run for use in subsequent runs. Through the use of the NOMACR parameter in BLOCK I or the LIB= MACRXS parameter in BLOCK III of the input, the user can easily suppress the re-execution of the working-cross-section-file submodule in subsequent code executions.

5. SOLVER-Input-File Submodule Execution Control. The SOLVER-input-file submodule, secondary overlay (1,8) shown in Fig. 2, processes the BLOCK V card-image input and creates the code-dependent interface file SOLINP for use by the SOLVER module.

This submodule will be executed and the SOLINP file created if both (i) the BLOCK I input parameter NOSLNP is set (or defaulted) to zero and (ii) BLOCK V card-image input is supplied.

The SOLVER-input-file submodule will not be executed (no SOLINP file created) if either (i) the BLOCK I input parameter NOSLNP is set to unity or (ii) all BLOCK V card-image input is omitted from the input "deck."

6. EDIT-Input-File Submodule Execution Control. The EDIT-input-file submodule, secondary overlay (1,9) shown in Fig. 2, processes the BLOCK VI card-image input and creates the code-dependent interface file EDITIT for use by the EDIT module of ONEDANT.

The EDIT-input-file submodule will be executed and the EDITIT file created if both (i) the BLOCK I input parameter NOEDTT is set (or defaulted) to zero and (ii) BLOCK VI card-image input is supplied.

This submodule will not be executed (no EDITIT file created) if either (i) the BLOCK I input parameter NOEDTT is set to unity or (ii) all BLOCK VI card-image input is omitted from the input "deck."

7. Adjoint-Reversal Submodule Execution Control. The adjoint-reversal submodule, secondary overlay (1,10) shown in Fig. 2, processes the MACRXS code-dependent cross-section interface file and creates the code-dependent interface file ADJMAC, the adjoint-reversed counterpart to the MACRXS file. This is described in Ch. V. K.

The adjoint-reversal submodule will be executed if both (i) the BLOCK I input parameter NOADJM is set (or defaulted) to zero and (ii) the BLOCK V input quantity ITH (found in the collective array SOLIN) is set to unity.

The submodule will not be executed (no ADJMAC file created) if either (i) the BLOCK I input parameter NOADJM is set to unity or (ii) the BLOCK V input quantity ITH (found in the collective array SOLIN) is set to zero.

IX. ERROR DIAGNOSTICS

A comprehensive error-checking capability has been provided in the ONEDANT code package. Most of the checks are in the INPUT module to ensure that the input data are correct, insofar as the code can determine, before execution of the problem commences. Other checks are made in the SOLVER and EDIT modules to ensure that the modules are executing the desired problem properly.

One important feature of the error diagnostics in the INPUT module is that an error will normally not cause an immediate termination of execution. Instead, the code will attempt to process remaining data in the offending input BLOCK and/or in remaining input BLOCKS. Once all remaining input has been processed (if possible) the run will be terminated.

Error messages are normally provided in at least two places in the output. The first error message is printed at the time that the error was detected by the code. Such messages will be imbedded in the printed output, but they are clearly marked for easy spotting. The second error message will normally occur in the RUN HIGHLIGHTS provided at the end of the printed output. These RUN HIGHLIGHTS provide a printed summary of the code package execution. The user is encouraged to always check the RUN HIGHLIGHTS following a run to quickly ascertain if the completed run did what it was supposed to.

A. Examples of Errors and Resulting Messages

Several examples of common input errors and the resulting error message printouts are provided below.

Sample Error 1. Unacceptable Input Parameter Value

ONEDANT uses the BLOCK I input parameter IGEOM to specify the geometry of the problem to be solved. Permissible values for IGEOM are 1 (plane geometry) 2 (cylindrical geometry), or 3 (spherical geometry). For this sample error a value of IGEOM= 4 has been entered. All other input data are correct. When the ONEDANT INPUT module tries to process the BLOCK V SOLVER module card-image input data in order to create the SOLINP interface file, it notes that IGEOM was set to an unacceptable value and prints the following fatal error message:

```
*****  
***ERROR** BAD IGEOM  
*****
```

The RUN HIGHLIGHTS, shown below, indicate that the SOLINP file could not be created (the double asterisks around the message denote a fatal error condition). A final fatal input error message to indicate termination of the run is also provided.

```

*****
*
*   ALL MODULES ARE TENTATIVELY GO.   *
*   INTERFACE FILE GEODST WRITTEN.   *
*   CROSS SECTIONS FROM CARDS.       *
*   INTERFACE MIXING FILES WRITTEN.  *
*   INTERFACE FILE ASGMAT WRITTEN    *
*   XS FILES MACRXS,SNXEOT WRITTEN.  *
* **UNABLE TO WRITE GOOD SOLINP FILE** *
*   EDIT MODULE EXECUTION SUPPRESSED. *
*   NEITHER EDITIT NOR EDIT CARDS EXIST.*
*       ** FATAL INPUT ERRORS **     *
*
*****

```

Sample Error 2. Misspelled Input Array Name

A common input error is that of misspelling the name of an input array. In this example, the BLOCK II card-image input array XMESH has been misspelled as XMESSH. The INPUT module is thus presented with an unrecognizable and undefined array name resulting in the following error message:

```

*ERRDR* CARD      14 *2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)2 4 6 8(6)2 4 6
                  XMESSH=0,28,34,62,90,100,XINTS=14,4,2R 14,8,ZONES=1,4,1,2,3 T

*ERROR* ARRAY NAME XMESSH ARRAY NUMBER      -1

*ERROR* COLUMNS  1 - 7

UNDEFINED ARRAY NAME

```

The first line of the error message indicates that an error was found on input card 14. This is followed by the card-image column numbers. Directly below this, the card-image is reproduced. The third line indicates that the array name XMESSH is in error and that this array has been given a number -1. (Acceptable arrays are given positive integer identification numbers by the code.) The next line says that the error occurred in columns 1 through 7 on the card-image. Finally, the message that the array name is undefined is provided.

The RUN HIGHLIGHTS indicate that the GEODST interface file has not been created due to the error in the array name. Note, however, that the HIGHLIGHTS

indicate that all remaining input data have been processed satisfactorily following the detection of the error and all other interface files were properly created before the run was terminated.

RUN HIGHLIGHTS

```

*****
*
*   ALL MODULES ARE TENTATIVELY GO.
* **UNABLE TO WRITE GOOD GEODST FILE**
*   CROSS SECTIONS FROM CARDS.
*   INTERFACE MIXING FILES WRITTEN.
*   INTERFACE FILE ASGMAT WRITTEN
*   XS FILES MACRXS.SNXEDT WRITTEN.
*   INTERFACE FILE SOLINP WRITTEN.
*   FDIT MODULE EXECUTION SUPPRESSED.
*   NEITHER EDITIT NOR EOIT CARDS EXIST.
*   ** FATAL INPUT ERRORS **
*
*****

```

Sample Error 3. Input Block Terminator Omitted

As described in Ch IV.A., each card-image input BLOCK must be terminated with a delimited T, the BLOCK terminator. In this example this terminal "T" has been omitted from the end of the BLOCK I card-image input. The offending portion actual card-image input for this case is shown below.

```

*****
*
*   ...LISTING OF CARDS IN THE INPUT STREAM.
*
*   1.      5      0      0
*   2. TEST CASE A - ALL CDS  ERROR CHECK  MISSING T BLOCK I
*   3. CROSS SECTIONS - 6E12 CARDS (DTF FORMAT)
*   4. GEOMETRY      - CARDS
*   5. MIXING        - CARDS
*   6. EDITS         - NONE
*   7. /
*   8. / ***** BLOCK I INPUT - MASTER CONTROLS *****
*   9. IGEOM=2, NGROUP=2, ISN=4, MT=4, NZONE=4, NISO=6, IM=5, IT=54
*  10. /
*  11. / ***** BLOCK II INPUT - GEOMETRY *****
*  12. / THIS GEOMETRY SPECIFIES A FAIRLY COARSE MESH
*  13. /      1. DELTA X APPROX. DNE MFP
*  14. XMesh=0,28,34,62,90,100,XINTS=14.4,2R 14.8,ZONES=1.4,1.2,3 T
*  15. /
*
*****

```

Note that there is no delimited T at the end of line 9.

As a result of this omission the following message is printed:

```
*****  
*KEY END   BLOCK I READ*  
*****
```

```
*****  
***ERROR** CURRENT BLOCK CONTAINS ARRAYS BELONGING TO OTHER BLOCKS  
*****
```

```
*  
* NO. OF   FROM  
* ARRAYS  BLOCK  
*  
*      8      I  
*      3      II  
*      0      III  
*      0      IV  
*      0      V  
*      0      VI
```

```
***ERROR** IN BLOCK IDENTIFICATION  
*****
```

The first line of the message indicates that the INPUT module has finished reading the card-image input that it thinks belongs in BLOCK I. (The code has actually read the 8 array entries on line 9 of the input but has continued reading until it found the terminal T following the three array entries belonging to BLOCK II on line 14 of the input.) The second line of the error message indicates that arrays that do not belong in BLOCK I have been found (the three arrays XMESH, XINTS, and ZONES). Next is printed a table indicating that eight arrays from BLOCK I and three from BLOCK II were discovered in the BLOCK I card-image reading process. The final error message of an error in BLOCK identification is self-explanatory.

It should be noted that when arrays from other BLOCKS are found in any given BLOCK, the code will terminate execution immediately and no RUN HIGHLIGHTS are provided.

Sample Error 4. Invalid Entry for BLOCK I Input Parameter.

As described in Ch. 1V, certain card-image input is always required in BLOCK I for a ONEDANT code execution. Specifically, the eight parameters in the collective array DIMENS, (ICEOM, NGROUP, ISN, NISO, MT, NZONE, IM, and IT) are required to be entered as positive integers. In this example, one of these parameters, IT, has been incorrectly entered with a value of zero. It should be noted that if one of these parameters is omitted altogether, the code will default its value to zero.

The code prints the following fatal error message:

```
.....  
***ERROR** BLOCK I ENTRY .LE. ZERO  
.....
```

• IT

The message is self-explanatory. No RUN HIGHLIGHTS are provided when BLOCK I input errors are encountered.

Sample Error 5. Incorrect Number of Entries in an Input Array.

Several input arrays available as input to ONEDANT require a predetermined number of entries. In this example the XINTS array in the BLOCK II card-image input was provided with only four entries instead of the five it should have had. The error message provided by ONEDANT is shown below.

```
*ERROR* CARD`      14 *2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6 8(5)2 4 6 8(6  
      XMesh=0.28,34,62,90,100,XINTS=14.4,2R 14,ZONES=1,4,1,2,3 T
```

```
*ERROR* ARRAY NAME XINTS  ARRAY NUMBER      4
```

```
ARRAY UNDERLOADED
```

```
COUNT LOADED= 4/COUNT DESIRED= 5
```

The first line indicates that an error was detected on input "card" number 14. This is followed by the individual column numbers for the card. The second line gives the actual card-image of card 14. Note that there are only four entries in the XINTS array. The next line points out that an error was detected in the XINTS array and that this array is assigned array number 4. The final two lines indicate that XINTS was underloaded with only four entries counted and loaded when the expected or desired number of entries should have been five.

In this case the remaining blocks of input data were successfully processed and the RUN HIGHLIGHTS are provided as shown below.

RUN HIGHLIGHTS

```

*****
*
*   ALL MODULES ARE TENTATIVELY GO.   *
* **UNABLE TO WRITE GOOD GEODST FILE** *
*   CROSS SECTIONS FROM CARDS.      *
*   INTERFACE MIXING FILES WRITTEN.  *
*   INTERFACE FILE ASGMAT WRITTEN    *
*   XS FILES MACRXS.SNXEDT WRITTEN.  *
*   INTERFACE FILE SOLINP WRITTEN.   *
*   EDIT MODULE EXECUTIDN SUPPRESSED.*
*   NEITHER EDITIT NOR EDIT CARDS EXIST.*
*       ** FATAL INPUT ERRORS **     *
*
*****

```

Note that the fatal XINTS error prevented the code from creating the necessary GEODST interface file. The run was thus terminated after the remaining input data were processed.

Sample Error 6. Misplaced Array Identifier.

As discussed in Chs. III.A.5., III.C.5, and IV.A, arrays in the card-image input are identified by a Hollerith name or a number immediately followed by an array identifier - an equals (=) sign, a dollar (\$) sign, or an asterisk (*). The array identifier is required so that the code can recognize that the array name or number is indeed an array name or number and not an ordinary data item. In this sample a blank was inadvertently placed between the array name (ASSIGN) and the array identifier (=). The resulting message is shown below.

```

*ERROR* CARD      63 *2 4 6 8(1)2 4 6 8
                  ASSIGN =MATLS, T

*ERROR* COLUMNS  8 - 8

**ERROR**BLANK PRECEDING AN ARRAY IDENTIFIER(=, $, OR *)

```

```

.....
***ERROR** READING MIXING
.....

```

The first line of the message indicates that an error was found on input card 63 and is followed by the card-image column numbers. Directly below this, the card-image is reproduced. The third line indicates that the error was detected in column 8 of the card-image and the fourth line provides the self-explanatory message that a blank was found preceding the array identifier. The final line is also self-explanatory.

B. Comments Regarding Multiple Errors

As a result of the INPUT module's attempt to continue processing card-image input after a fatal error has been detected, it is possible for multiple errors to be diagnosed and for multiple error messages to be printed.

When multiple error messages are printed, the user should check to see if one or more of the errors was due to a preceding error. In other words, a particular input error may cause a chain reaction of other errors. For example, suppose that the entry IT were inadvertently omitted from the BLOCK I input as in Sample Error 4. The code will thus record a value of IT= 0. An otherwise correct entry for the XINTS array in BLOCK II, however, will now appear incorrect to the code since the code checks to see if

$$\sum_{I=1}^{IM} XINTS(I) = IT \quad ,$$

and a message to the effect that the fine-mesh specifications (XINTS array) is not equal to IT will be printed. The user is thus advised to review multiple error messages starting with the first message printed in order to determine which errors are independent of other errors and which are results of a preceding error.

I. MACRXS File

The MACRXS code-dependent interface file is the working cross-section file for the SOLVER module. On the MACRXS file are the Material macroscopic cross sections arranged in energy-group order. The contents of this file are described below.

```
C*****
C                                     DATE 09/18/81
C
C   CF          MACRXS
C   CE          CODE DEPENDENT MACROSCOPIC MULTIGROUP CROSS SECTION FILE
C   CE          FOR USE IN ONEDANT SOLVER MOOULE
C
C*****
C
C   CN          THIS FILE PROVIDES A BASIC BROAD GROUP
C   CN          LIBRARY, ORDERED BY GROUP
C
C   C           ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY
C
C-----
C   CS          FILE STRUCTURE
C
C   CS          RECORD TYPE                PRESENT IF
C   CS          =====
C   CS          FILE IDENTIFICATION        ALWAYS
C   CS          FILE CONTROL               ALWAYS
C   CS          FILE DATA                 ALWAYS
C
C   CS          ***** (REPEAT FOR ALL GROUPS)
C   CS          *   PRINCIPAL CROSS SECTIONS        ALWAYS
C   CS          *   SCATTERING CONTROL DATA        NORD.NE.O
C   CS          *   SCATTERING MATRIX              NORD.NE.O
C   CS          *****
C
C-----
C   CR          FILE IDENTIFICATION
C
C   CL          HNAME,(HUSE(I),I=1,2),IVERS
C
C   CW          1+3*MULT=NUMBER OF WORDS
C
C   CD          HNAME          HOLLERITH FILE NAME - MACRXS - (A6)
C   CD          HUSE(1)        HOLLERITH USER IDENTIFICATION (A6)
C   CD          IVERS          FILE VERSION NUMBER
C   CD          MULT           DOUBLE PRECISION PARAMETER
C   CD          1- A6 WORD IS SINGLE WORD
C   CD          2- A6 WORD IS DOUBLE PRECISION WORD
C
C-----
```



```

C-----
CR          FILE CONTROL
C
CL  NGROUP,NMAT,NORD,NED, IDPF,LNG,MAXUP,MAXDN
C
CW  B=NUMBER OF WORDS
C
CD  NGROUP          NUMBER OF ENERGY GROUPS IN FILE
CD  NMAT            NUMBER OF MATERIALS IN FILE
CD  NORD            NUMBER OF LEGENDRE SCATTERING ORDERS
CD  NED             NUMBER OF EXTRA EDIT CROSS SECTIONS (IN ADDITION
CD                  TO THE 4 BASIC PRINCIPAL CROSS SECTIONS)
CD  IDPF            O/1 NO/YES CROSS SECTION DATA ARE DOUBLE PRECISION
CD  LNG             NUMBER OF THE LAST NEUTRON GROUP(FOR COUPLED SETS)
CD  MAXUP           MAXIMUM NUMBER OF UPSCATTER GROUPS
CD  MAXDN           MAXIMUM NUMBER OF DOWNSCATTER GRDUPS
C
C-----
C
C
C-----
CR          FILE DATA
C
CL  (HMAT(1),I=1,NMAT),(HED(J),J=1,NEDT),(VEL(N),N=1,NGROUP),
CL  1(EMAX(N),N=1,NGROUP),EMIN
C
CW  NMAT+NEDT+2*NGROUP+1=NUMBER OF WORDS
C
CD  HMAT(I)         HOLLERITH MATERIAL LABEL FOR MATERIAL I (A6)
CD  HED(J)          HOLLERITH LABEL FOR J-TH CROSS SECTION POSITION (A6-
CD  VEL(N)          MEAN NEUTRON VELOCITY IN GROUP N (CM/SEC)
CD  EMAX(N)         MAXIMUM ENERGY BDL'ND OF GROUP N (EV)
CD  EMIN            MINIMUM ENERGY BOUND OF SET (EV)
CD  NEDT           NED+4
C
CN                      THE FOUR BASIC PRINCIPAL CROSS SECTIONS
CN                      ALWAYS PRESENT ARE:
CN
CN                      HED(1) = 3HCHI
CN                      HED(2) = 6HNUSIGF
CN                      HED(3) = 5HTOTAL
CN                      HED(4) = 3HABS
C
C-----
C-----
CR          PRINCIPAL CROSS SECTIONS FOR GROUP N
C
CL  ((C(I,J),I=1,NMAT),J=1,NEDT)
C
CW  NMAT*NEDT=NUMBER OF WORDS
C
CD  C(I,J)          PRINCIPAL CROSS SECTIONS
C
CN                      BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:
CN
CN                      J=1  FISSION SPECTRUM
CN                      J=2  FISSION NU*FISSION CROSS SECTION
CN                      J=3  TOTAL CROSS SECTION
CN                      J=4  ABSORPTION CROSS SECTIOI:
C
C-----

```

```

C-----
CR          SCATTERING CONTROL BLOCK FOR GROUP N          -
C                                                    -
CC          PRESENT IF NORD.GT.O                          -
C                                                    -
CL          ((NGPB(L,J),L=1,NORD),J=1,NMAT)              -
CL          ((IFSG(L,J),L=1,NORD),J=1,NMAT)              -
C                                                    -
CW          2*NORD*NMAT=NUMBER OF WORDS                   -
C                                                    -
CD          NGPB(L,J)   NUMBER OF SOURCE GROUPS THAT CAN SCATTER INTO GROUP N -
CD          IFSG(L,J)   GROUP NUMBER OF THE FIRST SOURCE GROUP                -
CD          L           LEGENDRE ORDER NUMBER                    -
CD          J           MATERIAL NUMBER                          -
C                                                    -
C-----

C-----
CR          SCATTERING SUB-BLOCK FOR GROUP N          -
C                                                    -
CC          PRESENT IF NORD.GT.O                          -
C                                                    -
CL          (SCAT(I),I=1,NTAB)                            -
C                                                    -
CW          NTAB=NUMBER OF WORDS                           -
C                                                    -
CD          SCAT(I)     SCATTERING CROSS SECTION           -
C                                                    -
CD          NTAB        TABLE LENGTH OF THE CROSS SECTIONS FOR SCATTERING -
CD                      INTO GROUP N. THIS IS FOR ALL MATERIALS AND ALL      -
CD                      LEGENDRE ORDERS, THUS IT IS THE SUM OF NGPB(L,J)    -
CD                      OVER L FROM 1 TO NORD AND OVER J FROM 1 TO NMAT.    -
C                                                    -
CN          THE SCATTERING CROSS SECTIONS ARE PACKED IN BANDS, -
CN          ONE FOR EACH LEGENDRE ORDER AND MATERIAL. EACH BAND -
CN          CONTAINS THE NGPB GROUPS WHICH SCATTER INTO GROUP -
CN          N. THE FIRST SOURCE GROUP NUMBER IS IFSG AND -
CN          THE LAST IS IFSG-NGPB+1. THE NORD BANDS FOR THE -
CN          FIRST MATERIAL APPEAR FIRST (P0, P1, ...) FOLLOWED -
CN          BY THE NORD BANDS FOR THE SECONO, ETC.          -
C                                                    -
CN          HIGHER LEGENDRE ORDER SCATTERING CROSS SECTIONS -
CN          INCLUDE A 2*L+1 FACTOR WHERE L IS THE LEGENDRE -
CN          ORDER.                                          -
C                                                    -
C-----
CEDF

```

II. SNXEDT File

The SNXEDT file is the working cross-section file for the EDIT module. On the SNXEDT file are the Isotope microscopic cross sections arranged in energy-group order. Although the EDIT module will read any SNXEDT file constructed as described below, all SNXEDT files created by the INPUT module will have the parameter NORD set to zero so that scattering cross sections will not appear on the created SNXEDT file.

```

C.....
C                                     DATE 09/18/81
C
C
CF          SNXEDT
CE          CODE DEPENDENT MICROSCOPIC MULTIGROUP CROSS SECTION FILE
CE          FOR USE IN ONEDANT EDITS
C
C.....
C
C
CN          THIS FILE PROVIDES A BASIC BROAD GROUP
CN          LIBRARY, ORDERED BY GROUP
C
C-----
CS          FILE STRUCTURE
CS
CS          RECORD TYPE                                PRESENT IF
CS          =====                                =====
CS          FILE IDENTIFICATION                       ALWAYS
CS          FILE CONTROL                               ALWAYS
CS          FILE DATA                                 ALWAYS
CS
CS          ***** (REPEAT FOR ALL GROUPS)
CS          *          (GROUP 1 IS FIRST)
CS          *          PRINCIPAL CROSS SECTIONS        ALWAYS
CS          *          SCATTERING CONTROL DATA        NORD.NE.O
CS          *          SCATTERING MATRIX              NORD.NE.O
CS          *****
C
C-----
C
C
C-----
CR          FILE IDENTIFICATION
C
CL          HNAME,(HUSE(I),I=1,2),IVERS
C
CW          1+3*MULT=NUMBER OF WORDS
C
CD          HNAME          HOLLERITH FILE NAME - SNXEDT - (A6)
CD          HUSE(I)        HOLLERITH USER IDENTIFICATION (A6)
CD          IVERS          FILE VERSION NUMBER
CD          MULT           DOUBLE PRECISION PARAMETER
CD                          1- A6 WORD IS SINGLE WORD
CD                          2- A6 WORD IS DOUBLE PRECISION WORD
C
C-----
C
C-----
CR          FILE CONTROL
C
CL          NGROUP,NISO,NORD,NED,IDPF,LNG,MAXUP,MAXDN
C
CW          8=NUMBER OF WORDS
C
CD          NGROUP        NUMBER OF ENERGY GROUPS IN FILE
CD          NISO          NUMBER OF ISOTOPES IN FILE
CD          NORD          NUMBER OF LEGENDRE SCATTERING ORDERS
CD          NED           NUMBER OF EXTRA EDIT CROSS SECTIONS (IN ADDITION
CD                          TO THE 4 BASIC PRINCIPAL CROSS SECTIONS)
CD          IDPF          0/1 NO/YES CROSS SECTION DATA ARE DOUBLE PRECISION
CD          LNG           NUMBER OF THE LAST NEUTRON GROUP(FOR COUPLED SETS)
CD          MAXUP        MAXIMUM NUMBER OF UPSCATTER GROUPS
CD          MAXDN        MAXIMUM NUMBER OF DOWNSCATTER GROUPS
C
C-----

```

```

C-----
CR          FILE DATA
C
CL      (HISO(I),I=1,NISO),(HED(J),J=1,NEDT),(VEL(N),N=1,NGROUP),
CL      1(EMAX(N),N=1,NGROUP),EMIN
C
CW      NISO+NEDT+2*NGROUP+1=NUMBER OF WORDS
C
CD      HISO(I)          HOLLERITH ISOTOPE LABEL FOR ISOTOPE I (A6)
CD      HED(J)          HOLLERITH LABEL FOR EDIT NUMBER J (A6)
CD      VEL(N)          MEAN NEUTRON VELOCITY IN GROUP N (CM/SEC)
CD      EMAX(N)         MAXIMUM ENERGY BOUND OF GROUP N (EV)
CD      EMIN            MINIMUM ENERGY BOUND OF SET (EV)
CD      NEDT            NED+4
C
CN          THE FOUR BASIC PRINCIPAL CROSS SECTION EDITS
CN          ALWAYS PRESENT ARE:
CN
CN          HED(1) = 3HCHI
CN          HED(2) = 6HNUSIGF
CN          HED(3) = 5HTOTAL
CN          HED(4) = 3HABS
C
C-----

```

```

C-----
CR          PRINCIPAL CROSS SECTIONS FOR GROUP N
C
CL      ((C(I,J),I=1,NISO),J=1,NEDT)
C
CW      NISO*NEDT=NUMBER OF WORDS
C
CD      C(I,J)          PRINCIPAL CROSS SECTIONS
C
CN          BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:
CN
CN          J=1  FISSION SPECTRUM
CN          J=2  FISSION NU*FISSION CROSS SECTION
CN          J=3  TOTAL CROSS SECTION
CN          J=4  ABSORPTION CROSS SECTION
C
C-----

```

```

C-----
CR          SCATTERING CONTROL BLOCK FOR GROUP N
C
CC          PRESENT IF NORD.GT.O
C
CL      ((NGPB(L,J),L=1,NORO),J=1,NISO)
CL      ((IFSG(L,J),L=1,NORO),J=1,NISO)
C
CW      2*NORD*NISO=NUMBER OF WORDS
C
CD      NGPB(L,J)      NUMBER OF SOURCE GROUPS THAT CAN SCATTER INTO GROUP N
CD      IFSG(L,J)      GROUP NUMBER OF THE FIRST SOURCE GROUP
CD      L              LEGENDRE ORDER NUMBER
CD      J              ISOTOPE NUMBER
C
C-----

```

```

C-----
CR          SCATTERING SUB-BLOCK FOR GROUP N
C
C          PRESENT IF NORD.GT.O
C
CL      (SCAT(I),I=1,NTAB)
C
CW      NTAB=NUMBER OF WORDS
C
CD      SCAT(I)          SCATTERING CROSS SECTION
C
CD      NTAB            TABLE LENGTH OF THE CROSS SECTIONS FOR SCATTERING
CD                      INTO GROUP N. THIS IS FOR ALL ISOTOPES AND ALL
CD                      LEGENDRE ORDERS, THUS IT IS THE SUM OF NGPB(L,J)
CD                      OVER L FROM 1 TO NORD AND OVER J FROM 1 TO NISO.
C
CN                      THE SCATTERING CROSS SECTIONS ARE PACKED IN BANDS,
CN                      ONE FOR EACH LEGENDRE ORDER AND ISOTOPE. EACH BAND
CN                      CONTAINS THE NGPB GROUPS WHICH SCATTER INTO GROUP
CN                      N. THE FIRST SOURCE GROUP NUMBER IS IFSG AND
CN                      THE LAST IS IFSG-NGPB+1. THE NORD BANDS FOR THE
CN                      FIRST ISOTOPE APPEAR FIRST (PO, P1, ...) FOLLOWED
CN                      BY THE NORD BANDS FOR THE SECOND, ETC.
C
CN                      HIGHER LEGENDRE ORDER SCATTERING CROSS SECTIONS
CN                      INCLUDE A 2*L+1 FACTOR WHERE L IS THE LEGENDRE
CN                      ORDER.
C
C-----
CEOF

```

III. ADJMAC File

The ADJMAC file is the adjoint-reversed counterpart to the MACRXS interface file.

```

C*****
C          DATE 09/18/81
C
CF          ADJMAC
CE          CODE DEPENDENT MACROSCOPIC MULTIGROUP CROSS SECTION FILE
CE          USED IN ONEDANT SOLVER MODULE FOR ADJOINT CALCULATIONS
C
C-----
C          THIS FILE PROVIDES A BASIC BRADAD GROUP
CN          LIBRARY, ORDERED BY GROUP
C
C          ORDER OF GROUPS IS ACCORDING TO INCREASING ENERGY
C
C-----
CS          FILE STRUCTURE
CS
CS          RECORD TYPE          PRESENT IF
CS          =====
CS          FILE IDENTIFICATION  ALWAYS
CS          FILE CONTROL         ALWAYS
CS          FILE DATA           ALWAYS
CS
CS          ***** (REPEAT FOR ALL GROUPS)
CS          *          PRINCIPAL CROSS SECTIONS          ALWAYS
CS          *          SCATTERING CONTROL DATA          NORD.NE.O
CS          *          SCATTERING MATRIX                NORD.NE.O
CS          *****
C
C-----

```

```

C-----
CR          FILE IDENTIFICATION
C
CL      HNAME.(HUSE(I),I=1,2),IVERS
C
CW      1+3*MULT=NUMBER OF WORDS
C
CD      HNAME          HOLLERITH FILE NAME - ADJMAC - (A6)
CD      HUSE(I)        HOLLERITH USER IDENTIFICATION (A6)
CD      IVERS          FILE VERSION NUMBER
CD      MULT           DOUBLE PRECISION PARAMETER
CO              1- A6 WORD IS SINGLE WORD
CD              2- A6 WORD IS DOUBLE PRECISION WORD
C
C-----

```

```

C-----
CR          FILE CONTROL
C
CL      NGROUP,NMAT,NORD,NED,IDPF,LNG,MAXUP,MAXDN
C
CW      B=NUMBER OF WORDS
C
CD      NGROUP        NUMBER OF ENERGY GROUPS IN FILE
CD      NMAT          NUMBER OF MATERIALS IN FILE
CD      NORD          NUMBER OF LEGENDRE SCATTERING ORDERS
CD      NED           NUMBER OF EXTRA EDIT CROSS SECTIONS (IN ADDITION
CD                   TO THE 4 BASIC PRINCIPAL CROSS SECTIONS)
CD      IDPF         O/1 NO/YES CROSS SECTION OATA ARE DOUBLE PRECISION
CD      LNG          NUMBER OF THE LAST NEUTRON GROUP(FDR CDUPLD SETS)
CD      MAXUP        MAXIMUM NUMBER OF UPSCATTER GROUPS
CD      MAXDN        MAXIMUM NUMBER OF DDWNSCATTER GROUPS
C
C-----

```

```

C-----
CR          FILE DATA
C
CL      (HMAT(I),I=1,NMAT),(HED(J),J=1,NEDT),(VEL(N),N=1,NGROUP),
CL      1(EMAX(N),N=1,NGROUP),EMIN
C
CW      NMAT+NEDT+2*NGROUP+1=NUMBER OF WORDS
C
CD      HMAT(I)       HOLLERITH MATERIAL LABEL FOR MATERIAL I (A6)
CD      HED(J)        HOLLERITH LABEL FOR J-TH CRDSS SECTION POSITION (A6-
CD      VEL(N)        MEAN NEUTRON VELOCITY IN GROUP N (CM/SEC)
CD      EMAX(N)       MAXIMUM ENERGY BOUND OF GROUP N (EV)
CD      EMIN          MINIMUM ENERGY BOUND OF SET (EV)
CD      NEDT          NED+4
C
CN              THE FOUR BASIC PRINCIPAL CROSS SECTIONS
CN              ALWAYS PRESENT ARE:
CN
CN              HED(1) = 3HCHI
CN              HED(2) = 6HNUSIGF
CN              HED(3) = 5HTOTAL
CN              HED(4) = 3HABS
C
C-----

```

```

C-----
CR          PRINCIPAL CROSS SECTIONS FOR GROUP N
C
CL      ((C(I,J),I=1,NMAT),J=1,NEDT)
C
CW      NMAT*NEDT=NUMBER OF WORDS
C
CD      C(I,J)          PRINCIPAL CROSS SECTIONS
C
CN          BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:
CN
CN              J=1  FISSION SPECTRUM
CN              J=2  FISSION NU*FISSION CROSS SECTION
CN              J=3  TOTAL CROSS SECTION
CN              J=4  ABSORPTION CROSS SECTION
C
C-----
C
C-----
CR          SCATTERING CONTROL BLOCK FOR GROUP N
C
CC          PRESENT IF NORD.GT.O
C
CL      ((NGPB(L,J),L=1,NORD),J=1,NMAT)
CL      ((IFSG(L,J),L=1,NORD),J=1,NMAT)
C
CW      2*NORD*NMAT=NUMBER OF WORDS
C
CD      NGPB(L,J)      NUMBER OF SOURCE GROUPS THAT CAN SCATTER INTO GROUP N
CD      IFSG(L,J)      GROUP NUMBER OF THE FIRST SOURCE GROUP
CD      L              LEGENDRE ORDER NUMBER
CD      J              MATERIAL NUMBER
C
C-----

```

```

C-----
CR          SCATTERING SUB-BLOCK FOR GRDUP N
C
CC          PRESENT IF NDRD.GT.O
C
CL      (SCAT(I),I=1,NTAB)
C
CW      NTAB=NUMBER OF WORDS
C
CD      SCAT(I)        SCATTERING CROSS SECTION
C
CD      NTAB          TABLE LENGTH OF THE CROSS SECTIONS FOR SCATTERING
CD                    INTO GROUP N. THIS IS FOR ALL MATERIALS AND ALL
CD                    LEGENDRE ORDERS. THUS IT IS THE SUM OF NGPB(L,J)
CD                    OVER L FROM 1 TO NORD AND OVER J FROM 1 TO NMAT.
C
CN                    THE SCATTERING CROSS SECTIONS ARE PACKED IN BANDS,
CN                    ONE FOR EACH LEGENDRE ORDER AND MATERIAL. EACH BAND
CN                    CONTAINS THE NGPB GROUPS WHICH SCATTER INTO GROUP
CN                    N. THE FIRST SOURCE GROUP NUMBER IS IFSG AND
CN                    THE LAST IS IFSG-NGPB+1. THE NORD BANDS FOR THE
CN                    FIRST MATERIAL APPEAR FIRST (P0, P1, ...) FOLLOWED
CN                    BY THE NORD BANDS FOR THE SECOND, ETC.
C
CN                    HIGHER LEGENDRE ORDER SCATTERING CROSS SECTIONS
CN                    INCLUDE A 2*L+1 FACTOR WHERE L IS THE LEGENDRE
CN                    ORDER.
C
C-----
CEOF

```

IV. ASGMAT File

The ASGMAT interface file contains the information needed by the SOLVER and EDIT modules to assign materials to zones to create the zone macroscopic cross sections.

```
C*****-
C                                     DATE 09/18/81-
C-
C-
CF      ASGMAT-
CE      CODE DEPENDENT FILE ASSIGNING MATERIALS TO ZONES-
C-
C-
C*****-
C-
CN      THIS FILE CONTAINS THE INFORMATION FROM THE INPUT-
CN      ARRAYS ASSIGN= AND ASGMOD=-
C-
C-----
CS      FILE STRUCTURE-
CS-
CS      RECORD TYPE-                PRESENT IF-
CS      =====
CS      FILE IDENTIFICATION-        ALWAYS-
CS      FILE CONTROL-                ALWAYS-
CS      COMPATABILITY CODE-         ALWAYS-
CS      MATERIAL NAMES-              ALWAYS-
CS      ZONE NAMES-                  ALWAYS-
CS      NUMBER OF MATERIALS PER ZONE- MPZTOT.NE.O-
CS      MATERIAL LIST FOR ALL ZONES- MPZTOT.NE.O-
CS      MATERIAL CONCENTRATIONS-     MPZTOT.NE.O-
CS      MATERIAL CONCENTRATION FACTORS- MPZTOT.NE.O-
CS      CONCENTRATION MODIFIER-      MPZTOT.NE.O-
C-
C-----
C-
CR      FILE IDENTIFICATION-
C-
CL      HNAME,(HUSE(I),I=1,2),IVERS-
C-
CW      1+3*MULT=NUMBER OF WORDS-
C-
CD      HNAME      HOLLERITH FILE NAME - ASGMAT - (A6)-
CD      HUSE(1)    HOLLERITH USER IDENTIFICATION (A6)-
CD      IVERS      FILE VERSION NUMBER-
CD      MULT       DOUBLE PRECISION PARAMETER-
CD                  1- A6 WORD IS SINGLE WORD-
CD                  2- A6 WORD IS DOUBLE PRECISION WORD-
C-
C-----
```



```

C-----
CR          FILE CONTROL
C
CL      MT, NZONE, MPZTOT
C
CW      3=NUMBER OF WORDS
C
CD      1  MT          NUMBER OF MATERIALS
CD      2  NZONE       NUMBER OF ZONES
CD      3  MPZTOT      IN-SOLVER MIXING TABLE LENGTH
C
C-----
C
C-----
CR          COMPATABILITY CODE WORDS
C
CL      CODE1, CODE2
C
CW      2=NUMBER OF WORDS
C
CD      1  CODE1       DATE OF THE MACRXS FILE TO WHICH ASGMAT APPLIES
CD      2  CODE2       TIME OF THE MACRXS FILE TO WHICH ASGMAT APPLIES
C
C-----
C
C-----
CR          MATERIAL NAMES
C
CL      (MATNAM(I),I=1,MT)
C
CW      MT=NUMBER OF WORDS
C
CD      MATNAM(I) HOLLERITH NAME FOR THE I-TH MATERIAL
C
C-----
C
C-----
CR          ZONE NAMES
C
CL      (ZONNAM(I),I=1,NZONE)
C
CW      NZONE=NUMBER OF WORDS
C
CD      ZONNAM(I) HOLLERITH NAME FOR THE I-TH ZONE
C
C-----
C
C-----
CR          NUMBER OF MATERIALS PER ZONE
C
CC      PRESENT IF MPZTOT.NE.O
C
CL      (NUMZON(I),I=1,NZONE)
C
CW      NZONE=NUMBER OF WORDS
C
CD      NUMZON(I) NUMBER OF MATERIALS IN THE I-TH ZONE
C
C-----

```

```

C-----
CR          MATERIAL LIST FOR ALL ZONES          -
C          -                                     -
CC          PRESENT IF MPZTOT.NE.O              -
C          -                                     -
CL          (MATLST(I),I=1,MPZTOT)              -
C          -                                     -
CW          MPZTOT=NUMBER OF WORDS               -
C          -                                     -
CD          MATLST   PACKED LIST OF MATERIAL NUMBERS. MATERIALS FOR ZONE 1-
CD          FOLLOWED BY THOSE FOR ZONE 2, THEN ZONE3, ETC. -
C          -                                     -
C-----
C          -                                     -
CR          MATERIAL CONCENTRATIONS              -
C          -                                     -
CC          PRESENT IF MPZTOT.NE.O              -
C          -                                     -
CL          (CONC(I),I=1,MPZTOT)                -
C          -                                     -
CW          MPZTOT=NUMBER OF WORDS               -
C          -                                     -
CD          CONC(I)   CONCENTRATION OF THE I-TH MATERIAL IN THE MATLST
CD          ARRAY                                     -
C          -                                     -
C-----
C          -                                     -
CR          MATERIAL CONCENTRATIDN FACTORS        -
C          -                                     -
CC          PRESENT IF MPZTOT.NE.O              -
C          -                                     -
CL          (C1(I),I=1,MPZTOT)                  -
C          -                                     -
CW          MPZTOT=NUMBER OF WORDS               -
C          -                                     -
CD          C1(I)    CONCENTRATION FACTOR FOR THE I-TH MATERIAL IN THE
CD          MATLST ARRAY                             -
C          -                                     -
C-----
C          -                                     -
C          -                                     -
CR          CONCENTRATION MODIFIER                -
C          -                                     -
CC          PRESENT IF MPZTOT.NE.O              -
C          -                                     -
CL          CMDD                                  -
C          -                                     -
CW          1=NUMBER OF WORDS                    -
C          -                                     -
CD          CMOD    INPUT VALUE OF THE CONCENTRATION MODIFIER -
C          -                                     -
C-----
CEOF

```

V. SOLINP File

The SOLINP code-dependent interface file contains information specific to the SOLVER module, mainly the information from BLOCK V of the card-image input.

```

C-----
C                                     DATE 09/18/81
C
C      SOLINP
CF      CODE DEPENDENT FILE OF INFORMATION SPECIFIC TO THE
CE      DNEDANT SOLVER MODULE
C
C-----
C
C      FILE STRUCTURE
CS
CS      RECORD TYPE                                PRESENT IF
CS      =====
CS      FILE IDENTIFICATION                        ALWAYS
CS      TITLE CARD COUNT                          ALWAYS
CS      ***** (REPEAT FOR UP TO 10 CDS)
CS      *      TITLE CARD                          NHEAD.GT.0
CS      *****
CS      DIMENSION                                  ALWAYS
CS      RAW CONTROLS AND DIMENSIONS                ALWAYS
CS      RAW FLOATING INPUT DATA                   ALWAYS
CS      DEFAULTED CONTROLS AND DIMENSIONS          ALWAYS
CS      DEFAULTED FLOATING INPUT DATA             ALWAYS
CS      BNDRY TRANSFER FIRST SOURCE GROUP          IBL.EQ.4.OR.IBR.EQ.4
CS      BNDRY TRANSFER VECTOR LENGTHS             IBL.EQ.4.OR.IBR.EQ.4
CS      ***** (REPEAT FOR ALL GROUPS)
CS      *      LEFT BDRY TRANSFER VECTOR            IBL.EQ.4
CS      *      RIGHT BDRY TRANSFER VECTOR           IBR.EQ.4
CS      *****
CS      FINE MESH DENSITY FACTORS                   IDEN.EQ.1
CS      RADIUS MODIFIERS                            IEVT.EQ.4
CS      LEFT ALBEDOES                               IBL.EQ.4
CS      RIGHT ALBEDOES                              IBR.EQ.4
CS      SINGLE CHI ARRAY(FISSION SPECTRA)          INCHI.EQ.1
CS      ***** (REPEAT FOR ALL ZONES)
CS      *      CHI ARRAY(FISSION SPECTRA)          INCHI.EQ.2
CS      *****
CS      QUADRATURE WEIGHTS                          IQUAD.EQ.3
CS      QUADRATURE COSINES                          IQUAD.EQ.3
CS      ***** (REPEAT FOR ALL MOMENTS)
CS      *      SOURCE SPECTRUM                      IQOPT.EQ.1
CS      *      SOURCE SPATIAL DISTRIBUTION          IQOPT.EQ.2
CS      * ***** (REPEAT FOR ALL GROUPS)
CS      *      SOURCE SPATIAL DISTRIBUTION          IQOPT.EQ.3
CS      * *****
CS      *      SOURCE SPECTRUM                      IQOPT.EQ.4
CS      *      SOURCE SPATIAL DISTRIBUTION          IQOPT.EQ.4
CS      *****
CS      LEFT BOUNDARY ISOTROPIC SOURCES             IQL.EQ.-1
CS      RIGHT BOUNDARY ISOTROPIC SOURCES           IQR.EQ.-1
CS      ***** (REPEAT FOR ALL GROUPS)
CS      *      LEFT BOUNDARY ANISOTROPIC SOURCE    IQL.EQ.+1
CS      *      RIGHT BOUNDARY ANISOTROPIC SOURCE   IQR.EQ.+1
CS      *****
C-----

```

```

C-----
CR          FILE IDENTIFICATION
C
CL      HNAME.(HUSE(I),I=1,2).IVERS
C
CW      1+3*MULT=NUMBER OF WORDS
C
CD      HNAME          HOLLERITH FILE NAME - SOLINP - (A6)
CD      HUSE(I)        HOLLERITH USER IDENTIFICATION (A6)
CD      IVERS          FILE VERSION NUMBER
CD      MULT           DOUBLE PRECISION PARAMETER
CD                      1- A6 WORD IS SINGLE WORD
CD                      2- A6 WORD IS DOUBLL PRECISION WORD
C-----
C
C-----
CR          TITLE CARD COUNT
C
CL      NHEAD
C
CW      1=NUMBER OF WORDS
C
CD      NHEAD          NUMBER OF TITLE CARDS TO FOLLOW
C-----
C
C-----
CR          TITLE CARD
C
CC      PRESENT IF NHEAD.GT.O
C
CL      (TITLE(I),I=1,12)
C
CW      12=NUMBER OF A6 WORDS
C-----
C
C-----
CR          SPATIAL DIMENSION
C
CL      IDIMEN
C
CW      1=NUMBER OF WORDS
C
CD      IDIMEN=1 FOR DNEDANT
C-----

```

```

C-----
CR      RAW CONTROLS AND DIMENSIONS
C
CL      IEVT, ITH, ISCT, ISN, IQUAD, ISTART, ICSM, INCHI, IBL, IBR,
CL 1    IDEN, IPVT, I2ANG, IQOPT, IQAN, IQL, IQR, OITM, IITL, IITM,
CL 2    ITLIM, I1, FLUXP, XSECTP, FISSRP, SOURCP, GEOMP, ANGP, IACC, DUM1,
CL 3    DUM2, IAFLUX, ISBEDO, IBALP, DUM3, IBB, IBT, IITLD, IOT, IQB,
CL 4    IXM, IYM, IZM, IDENY, IDENZ
C
CW      45=NUMBER OF WORDS
C
CD 1    IEVT      TYPE OF CALCULATION
CD 2    ITH       0/1 - DIRECT/ADJOINT CALCULATION
CD 3    ISCT      LEGENDRE ORDER OF SCATTERING
CD 4    ISN       ANGULAR QUADRATURE ORDER
CD 5    IQUAD     SOURCE OF QUADRATURE SET
CD 6    ISTART   FLUX GUESS FLAG (ZERO FOR ONEDANT)
CD 7    ICSM     0/1 - NO/YES IN-SOLVER MIXING(FROM ASSIGN= )
CD 8    INCHI    0/1/2 - NONE/ONE CHI/ZONWISE CHI
CD 9    IBL      0/1/2/3/4 - LEFT BDRY CONDITION
CD 10   IBR      0/1/2/3/4 - RIGHT BDRY CONDITION
CD
CD 11   IDEN     0/1 - NO/YES FINE MESH DENSITY FACTORS
CD 12   IPVT     0/1/2 - NONE/K-EFF/ALPHA PARAMETRIC EIGENVALUE TYPE
CD 13   I2ANG    0/1 - NO/YES DD 2 ANGLE SLAB CALCULATION
CD 14   IQOPT    0/1/2/3/4/5 - INHOMOGENEOUS SOURCE OPTION
CD 15   IQAN     INHOMOGENEOUS SOURCE LEGENDRE ORDER
CD 16   IQL     -1/0/1 - ISOTROPIC/NONE/ANGULAR LEFT BOUNDARY SOURCE
CD 17   IQR     -1/0/1 - ISOTROPIC/NONE/ANGULAR RIGHT BOUNDARY SOURCE
CD 18   OITM     OUTER ITERATION LIMIT
CD 19   IITL     EARLY INNER ITERATION LIMIT
CD 20   IITM     NEAR CONVERGENCE INNER ITERATION LIMIT
CD
CD 21   ITLIM   TIME LIMIT IN SECONDS
CD 22   I1      NOT USED
CD 23   FLUXP   0/1/2 - NONE/ISOTROPIC/ALL MOMENTS FLUX PRINT
CD 24   XSECTP 0/1/2 - NONE/PRINCIPAL/ALL CROSS SECTION PRINT
CD 25   FISSRP  0/1 - NO/YES FISSION RATE PRINT
CD 26   SOURCP  0/1/2/3 - NO/AS READ/NORMALIZED/BOTH SOURCE PRINT
CD 27   GEOMP   0/1 - NO/YES FINE MESH GEOMETRY PRINT
CD 28   ANGP    0/1 - NO/YES ANGULAR FLUX PRINT
CD 29   IACC    ACCELERATION TYPE (FIXED AT 2 FOR ONEDANT)
CD 30   DUM1    NOT USED
CD 31   DUM2    NOT USED
CD 32   RAFLUX  0/1 - NO/YES WRITE ANGULAR FLUX FILE RAFLUX
CD 33   ISBEDO  ALBEDO PRESENCE CODE
CD 34   IBALP   0/1 - NO/YES PRINT BALANCES BY ZONE
CD 35   DUM3    NOT USED
CD 36   IBB     NOT USED BY ONEDANT
CD 37   IBT     NOT USED BY ONEDANT
CD 38   IITLD   NOT USED BY ONEDANT
CD 39   IOT     NOT USED BY ONEDANT
CD 40   IQB     NOT USED BY ONEDANT
CD
CD 41   IXM     NOT USED BY ONEDANT
CD 42   IYM     NOT USED BY ONEDANT
CD 43   IZM     NOT USED BY ONEDANT
CD 44   IDENY   NOT USED BY ONEDANT
CD 45   IDENZ   NOT USED BY ONEDANT
C-----

```

```

C-----
CR          RAW FLOATING DATA
C
CL      EV,  NORM, EPSO, EPSI, BHGT, BWTH, EVM, PV, XLAL, XLAH,
CL      XLAX, POD, EPSR, EPSX, EPST
C
CW      15=NUMBER OF WORDS
C
CD      1  EV          EIGENVALUE GUESS
CD      2  NORM        NORMALIZATION CONSTANT
CD      3  EPSO        OUTER ITERATION CONVERGENCE CRITERION
CD      4  EPSI        INNER ITERATION CONVERGENCE CRITERION
CD      5  BHGT        BUCKLING HEIGHT
CD      6  BWTH        BUCKLING WIDTH
CD      7  EVM         EIGENVALUE MODIFIER
CD      8  PV          PARAMETRIC VALUE
CD      9  XLAL        LAMBDA LOWER LIMIT FOR SEARCHES
CD     10  XLAH        LAMBDA UPPER LIMIT FOR SEARCHES
CD
CD     11  XLAX        SEARCH CONVERGENCE CRITERION
CD     12  PDD         PARAMETER OSCILLATION DAMPER
CD     13  EPSR        DIFFUSION PERIODIC BDRY ITERATION CONV. CRITERION
CD     14  EPSX        MAX FRACTIONAL POINTWISE CHANGE CRITERION
CD     15  EPST        NOT USED BY ONEDANT
C-----

C-----
CR          DEFAULTED CONTROLS AND DIMENSIONS
C
CN      THIS RECORD IS THE SAME FORMAT AS THE RAW CONTROLS AND DIMENSION
CN      RECORD ABOVE, BUT IT CONTAINS THE DEFAULTED VALUES FOR EACH
CN      VARIABLE
C-----

C-----
CR          DEFAULTED FLOATING DATA
C
CN      THIS RECORD IS THE SAME FORMAT AS THE RAW FLOATING DATA
CN      RECORD ABOVE, BUT IT CONTAINS THE DEFAULTED VALUES FOR EACH
CN      VARIABLE
C-----

C-----
CR          BOUNDARY TRANSFER FIRST SOURCE GROUP
C
CC      PRESENT IF IBL.EQ.4 .OR. IBR.EQ.4
C
CL      (IFSG(L),L=1,NGROUP),(IFSGR(R),R=1,NGROUP)
C
CW      2*NGROUP=NUMBER OF WORDS
C
CD      IFSGL        FIRST SOURCE GROUP FOR LEFT BOUNDARY
CD      IFSGR        FIRST SOURCE GROUP FOR RIGHT BOUNDARY
CD      NGRUP        NUMBER OF ENERGY GROUPS
C-----

C-----
CR          BOUNDARY TRANSFER VECTOR LENGTHS
C
CC      PRESENT IF IBL.EQ.4 .OR. IBR.EQ.4
C
CL      (LENL(N),N=1,NGROUP),(LENR(N),N=1,NGROUP)
C
CW      2*NGROUP=NUMBER OF WORDS
C
CD      LENL         LENGTH OF THE VECTOR OF TRANSFERS INTO GRP N
CD                   AT THE LEFT BOUNDARY
CD      LENR         LENGTH OF THE VECTOR OF TRANSFERS INTO GRP N
CD                   AT THE RIGHT BOUNDARY
C-----

```

```

C-----
CR      LEFT BOUNDARY TRANSFER VECTOR
C
CC      PRESENT IF IBL.EQ.4
C
CL      (TRL(I),I=1,NW)
C
CW      NW=NUMBER OF WORDS
C
CD      TRL(I)  I-TH TRANSFER INTO GROUP N FOR LEFT BOUNDARY
CD              THE FIRST VALUE IS THE TRANSFER FROM GROUP IFSG(LN)
CD              INTO GROUP N, NEXT IS FROM GROUP IFSG(LN)-1 TO N, ETC.
CD      NW      LENL(N)
CD      N        NUMBER OF THE RECEIVING GROUP
C-----
C
C-----
CR      RIGHT BOUNDARY TRANSFER VECTOR
C
CC      PRESENT IF IBR.EQ.4
C
CL      (TRR(I),I=1,NW)
C
CW      NW=NUMBER OF WORDS
C
CD      TRR(I)  I-TH TRANSFER INTO GROUP N FOR RIGHT BOUNDARY
CD              THE FIRST VALUE IS THE TRANSFER FROM GROUP IFSGR(N)
CD              INTO GROUP N, NEXT IS FROM GROUP IFSGR(N)-1 TO N, ETC.
CD      NW      LENR(N)
CD      N        NUMBER OF THE RECEIVING GROUP
C-----
C
C-----
CR      FINE MESH DENSITY FACTORS
C
CC      PRESENT IF IDEN.EQ.1
C
CL      (DEN(I),I=1,IT)
C
CW      IT=NUMBER OF WORDS
C
CD      IT      NUMBER OF FINE MESH INTERVALS
C-----
C
C-----
CR      RADIAL MODIFIERS
C
CC      PRESENT IF IEVT.EQ.4
C
CL      (RM(I),I=1,IM)
C
CW      IM=NUMBER OF WORDS
C
CD      IM      NUMBER OF COARSE MESH INTERVALS
C-----
C
C-----
CR      LEFT ALBEDOES
C
CC      PRESENT IF IBL.EQ.4
C
CL      (LBEDO(N),N=1,NGROUP)
C
CW      NGROUP=NUMBER OF WORDS
C-----

```

```

C-----
CR      RIGHT ALBEDOES
C
CC      PRESENT IF IBR.EQ.4
C
CL      (RBEDO(N),N=1,NGROUP)
C
CW      NGROUP=NUMBER OF WORDS
C-----
C
C-----
CR      CHI - FISSION SPECTRA
C
CC      PRESENT IF INCHI.NE.0
C
CL      (CHI(N),N=1,NGROUP)
C
CW      NGROUP=NUMBER OF WORDS
C-----
C
C-----
CR      QUADRATURE WEIGHTS
C
CC      PRESENT IF IQUAD.EQ.3
C
CL      (WGT(M),M=1,MM)
C
CW      MM=NUMBER OF WORDS
C
CD      MM      NUMBER OF ANGLES IN QUADRATURE SET
C-----
C
C-----
CR      QUADRATURE COSINES
C
CC      PRESENT IF IQUAD.EQ.3
C
CL      (MU(M),M=1,MM)
C
CW      MM=NUMBER OF WORDS
C-----
C
C-----
CR      SOURCE SPECTRUM
C
CC      PRESENT IF IQOPT.EQ.1 .OR. IQOPT.EQ.4
C
CL      (SOURCE(N),N=1,NGROUP)
C
CW      NGROUP=NUMBER OF WORDS
C-----
C
C-----
CR      SOURCE SPATIAL DISTRIBUTION
C
CC      PRESENT IF IQOPT.EQ.2 .OR. IQOPT.EQ.3 .OR. IQOPT.EQ.4
C
CL      (SOURCX(I),I=1,IT)
C
CW      IT=NUMBER OF WORDS
C-----

```



```

C-----
CR          LEFT BOUNDARY ISOTROPIC SOURCE      -
C                                                  -
CC          PRESENT IF IBL.EQ.-1                -
C                                                  -
CL          (SILEFT(N),N=1,NGROUP)              -
C                                                  -
CW          NGROUP=NUMBER OF WORDS              -
C-----
C-----
C-----
CR          RIGHT BOUNDARY ISOTROPIC SOURCE     -
C                                                  -
CC          PRESENT IF IBL.EQ.-1                -
C                                                  -
CL          (SIRITE(N),N=1,NGROUP)              -
C                                                  -
CW          NGROUP=NUMBER OF WORDS              -
C-----
C-----
C-----
CR          LEFT BOUNDARY ANISOTROPIC SOURCE    -
C                                                  -
CC          PRESENT IF IBL.EQ.+1                -
C                                                  -
CL          (SALEFT(M),M=1,MMHALF)              -
C                                                  -
CW          MMHALF=NUMBER OF WORDS              -
C-----
CD          SALEFT  LEFT BOUNDARY SOURCE FOR GRU -
CD          MMHALF  NUMBER OF ANGLES IN THE SOURCE(=MM/2)
C-----
C-----
CR          RIGHT BOUNDARY ANISOTROPIC SOURCE    -
C                                                  -
CC          PRESENT IF IBL.EQ.+1                -
C                                                  -
CL          (SARITE(M),M=1,MMHALF)              -
C                                                  -
CW          MMHALF=NUMBER OF WORDS              -
C-----
CD          SARITE  RIGHT BOUNDARY SOURCE FOR GR -
CD          MMHALF  NUMBER OF ANGLES IN THE SOURCE(=MM/2)
C-----
CEOF

```

VI. EDITIT File

The EDITIT code-dependent interface file contains information specific to the EDIT module, mainly information from BLOCK VI of the card-image input.

```

C.....
C                                     DATE 09/18/81
C
C
CF          EDITIT
CE          CODE DEPENDENT FILE OF INFORMATION SPECIFIC TO THE
CE          ONEDANT EDIT MOOULE
C
C
C.....
C
CN          THIS FILE CONTAINS THE CARD INPUT INFORMATION
CN          FROM BLOCK VI
C
C-----
CS          FILE STRUCTURE
C
CS          RECORD TYPE                PRESENT IF
CS          ==== =====
CS          FILE IDENTIFICATION        ALWAYS
CS          RAW CONTROLS AND DIMENSIONS ALWAYS
CS          DEFAULTED CONTROLS AND DIMENSIONS ALWAYS
CS          FINE GROUPS PER BROAD GROUP ALWAYS
CS          ZONE NUMBERS                NZNS.NE.O
CS          POINT NUMBERS              NIPE.NE.O
CS          CROSS SECTION POSITIONS    NPOS.NE.O
CS          ISOTOPE NUMBERS TO EDIT    NISO.NE.O
CS          MATERIAL NUMBERS TO EDIT   MACRO.NE.O
CS          CONSTITUENT NUMBERS TO EDIT NCONS.NE.O
CS          RESPONSE FUNCTION NAMES     IDOSE.NE.O
CS          *****
CS          ***** (REPEAT FOR ALL RESPONSE FUNCTIONS)
CS          * RESPONSE FUNCTION ENERGY VECTOR IDOSE.NE.O
CS          * RESPONSE FUNCTION SPATIAL VECTOR IDOSE.NE.O
CS          *****
CS          CROSS SECTION SUMMING ARRAY - IXSUM.NE.O
CS          RESPONSE FUNCTION SUMMING ARRAY IRSUM.NE.O
CS          FINE MESH DENSITY FACTORS   IDEN.NE.O
C
C-----
C
CR          FILE IDENTIFICATION
C
CL          HNAME,(HUSE(I),I=1,2),IVERS
C
CW          1+3*MULT=NUMBER OF WORDS
C
CD          HNAME          HOLLERITH FILE NAME - EDITIT - (A6)
CD          HUSE(I)        HOLLERITH USER IDENTIFICATION (A6)
CD          IVERS          FILE VERSION NUMBER
CD          MULT           DOUBLE PRECISION PARAMETER
CD                          1- A6 WORD IS SINGLE WORD
CD                          2- A6 WORD IS DDUBLE PRECISION WORD
C
C-----

```

```

-----
C
CR          RAW CONTROLS AND DIMENSIONS
C
CL  IEDOPT,PTED, NIPE, IKND, ZNED, NZNS,IXSUM IRSUM, NPDS, NISO,
CL  1 NCONS, MACRO, IDOSE, IGRPED, LNG, NBG, IDEN, JGROUP,AJED,
C
CW          25=NUMBER OF WORDS
C
CD  1 IEDOPT   NOT USED
CD  2 PTED    O/1 - NO/YES DO POINT EDIT
CD  3 NIPE    NUMBER OF POINTS TO EDIT
CD  4 BYVOLP  O/1 - NO/YES MULTIPLY REACTION RATES BY MESH VOLUME
CD  5 ZNED    O/1 - NO/YES DO ZONE EDIT
CD  6 NZNS    NUMBER OF EDIT ZONES
CD  7 IXSUM   LENGTH OF CROSS SECTION SUMMING TABLE
CD  8 IRSUM   LENGTH OF RESPONSE FUNCTION SUMMING TABLE
CD  9 NPOS    NUMBER OF CROSS SECTION POSITIONS TO EDIT
CD 10 NISO    NUMBER OF ISOTOPES TO EDIT
CL'
CD 11 NCONS   NUMBER OF ISOTOPES TO EDIT AS CONSTITUENTS
CD 12 MACRO   NUMBER OF MATERIALS TO EDIT
CD 13 IDOSE   NUMBER OF RESPONSE FUNCTIONS TO EDIT
CD 14 IGRPED  O/1/2/3 - ENERGY GROUP PRINT OPTIONS
CD 15 LNG     NUMBER OF THE LAST NEUTRON GROUP
CD 16 NBG     NUMBER OF BROAD ENERGY GROUPS
CD 17 IDEN    O/1 - NO/YES THERE ARE FINE MESH DENSITY FACTORS
CD 18 NGROUP  NUMBER OF FINE ENERGY GROUPS
CD 19 AJED    O/1 - NO/YES THIS IS AN ADJOINT EDIT
CD 20-27     NOT USED
C
-----
C
CR          DEFAULTED CONTROLS AND DIMENSIONS
C
CN  THIS RECORD IS THE SAME FORMAT AS THE RAW CONTROLS AND
CN  DIMENSION RECORD ABOVE, BUT IT CONTAINS THE DEFAULTED
CN  VALUES FOR EACH VARIABLE
C
-----
C
CR          FINE GROUPS PER BROAD GROUP
C
CL  (ICOLL(G),G=1,NBG)
C
CW  NBG=NUMBER OF WORDS
C
CD  ICOLL(G) NUMBER OF FINE GROUPS IN BROAD GROUP G
C
-----
C
CR          ZONE NUMBERS
C
CC  PRESENT IF NZNS.GT.0
C
CL  (EDZONE(I),I=1,IT)
C
CW  IT=NUMBER OF WORDS
C
CD  EDZONE(I) EDIT ZONE NUMBER FOR THE I-TH FINE MESH
C
-----

```

```

C-----
CR          POINTS TO EDIT
C
CC          PRESENT IF NIPE.GT.O
C
CL          (POINTS(I),I=1,NIPE)
C
CW          NIPE=NUMBER OF WORDS
C
CD          POINTS(I) NUMBER OF THE I-TH POINT TO EDIT
C
C-----
C
C-----
CR          CROSS SECTION POSITIONS TO EDIT
C
CC          PRESENT IF NPOS.GT.O
C
CL          (EDXS(I),I=1,NPOS)
C
CW          NPOS=NUMBER OF WORDS
C
CD          EDXS(I) POSITION NUMBER TO EDIT(IN NUMERIC FORM)
C
C-----
C
C-----
CR          ISOTOPE NUMBERS TO EDIT
C
CC          PRESENT IF NISO.GT.O
C
CL          (EDISOS(I),I=1,NISO)
C
CW          NISO=NUMBER OF WORDS
C
CD          EDISOS(I) ISOTOPE NUMBER TO EDIT(IN NUMERIC FORM)
C
C-----
C
C-----
CR          MATERIAL NUMBERS TO EDIT
C
CC          PRESENT IF MACRO.GT.O
C
CL          (EDMATS(I),I=1,MACRD)
C
CW          MACRO=NUMBER OF WORDS
C
CD          EDMATS(I) MATERIAL NUMBER TO EDIT(IN NUMERIC FORM)
C
C-----
C
C-----
CR          CONSTITUENT NUMBERS TO EDIT
C
CC          PRESENT IF NCONS.GT.O
C
CL          (EDCONS(I),I=1,NCONS)
C
CW          NCONS=NUMBER OF WORDS
C
CD          EDCONS(I) CONSTITUENT NUMBER TO EDIT(IN NUMERIC FORM)
C
C-----

```

```

C-----
CR      RESPONSE FUNCTION NAMES
C
CC      PRESENT IF IDOSE.GT.O
C
CL      (RSFNAM(I),I=1,IDOSE)
C
CW      IDOSE=NUMBER OF WORDS
C
CD      RSFNAM(I) HOLLERITH NAME FOR THE I-TH RESPONSE FUNCTION
C
C-----
C
C-----
CR      RESPONSE FUNCTION ENERGY VECTOR
C
CC      PRESENT IF IDOSE.GT.O
C
CL      (RSFE(I),I=1,NGROUP)
C
CW      NGROUP=NUMBER OF WORDS
C
CD      RSFE(I)  RESPONSE FOR GROUP I
C
C-----
C
C-----
CR      RESPONSE FUNCTION SPATIAL VECTOR
C
CC      PRESENT IF IDOSE.GT.O
C
CL      (RSFX(I),I=1,IT)
C
CW      IT=NUMBER OF WORDS
C
CD      RSFX(I)  RESPONSE FUNCTION FOR FINE MESH I
C
C-----
C
C-----
CR      CROSS SECTION SUMMING ARRAY
C
CC      PRESENT IF IXSUM.GT.O
C
CL      (MICSUM(I),I=1,IXSUM)
C
CW      IXSUM=NUMBER OF WORDS
C
CD      MICSUM  INPUT SUMMING ARRAY IN NUMERIC FORM
C
C-----
C
C-----
CR      RESPONSE FUNCTION SUMMING ARRAY
C
CC      PRESENT IF IRSUM.GT.O
C
CL      (IRSUMS(I),I=1,IRSUM)
C
CW      IRSUM=NUMBER OF WORDS
C
CD      IRSUMS  INPUT SUMMING ARRAY IN NUMERIC FORM
C
C-----
C
C-----
CR      FINE MESH DENSITY ARRAY
C
CC      PRESENT IF IOEN.GT.O
C
CL      (XDF(I),I=1,IT)
C
CW      IT=NUMBER OF WORDS
C
CD      XDF(I)  DENSITY FACTOR FOR THE I-TH FINE MESH
C
C-----
CEOF

```

VII. BXSLIB File

The BXSLIB code-dependent file contains, in binary form, the cross sections originally provided in BCD card-image form. It is thus essentially equivalent to the BCD card-image cross-section library and, as such, is not truly an interface file. Instead, it is more of a convenience file as described in Ch. V. C. 3. Nevertheless, as a code-dependent binary file, its file description is included in this appendix.

```

C*****
C                               DATE 09/18/81
C
CF          BXSLIB
CE          MICROSCOPIC GROUP NEUTRON CROSS SECTIONS FROM CARDS
C
CN          THIS FILE CONTAINS IN BINARY FORM THE
CN          BLOCK III ONEDANT INPUT TOGETHER WITH THE
CN          CROSS SECTIONS FROM THE ORIGINAL CARD LIBRARY.
C
C*****
C-----
C          FILE STRUCTURE
CS
CS          RECORD TYPE                PRESENT IF
CS          =====
CS          FILE IDENTIFICATION        ALWAYS
CS          FILE CONTROL                ALWAYS
CS          FILE DATA                  ALWAYS
CS
CS          ***** (REPEAT FOR ALL ISOTOPES)
CS          * ***** (REPEAT FOR ALL LEGENDRE ORDERS)
CS          * *      CROSS SECTION SET      ALWAYS
CS          * *****
CS          *****
C
C-----
C          FILE IDENTIFICATION
CR
CL          HNAME,(HUSE(I),I=1,2),IVERS
C
CW          1+3*MULT=NUMBER OF WORDS
C
C
CD          HNAME          HOLLERITH FILE NAME - BXSLIB - (A6)
CD          HUSE(1)        HOLLERITH USER IDENTIFICATION (A6)
CD          IVERS          FILE VERSION NUMBER
CD          MULT           DOUBLE PRECISION PARAMETER
CD                          1- A6 WORD IS SINGLE WORD
CD                          2- A6 WORD IS DOUBLE PRECISION WORD
C
C-----

```

APPENDIX B

ONEDANT SAMPLE PROBLEMS

In this appendix are presented the printed output from two sample problems. The first sample problem is a standard k_{eff} calculation with all input by means of card-images. The second sample problem is an edit-only problem in which edits are performed using the scalar fluxes and cross sections from the first sample problem.

I. Sample Problem 1: Standard k_{eff} Calculation

Sample Problem 1 is a standard k_{eff} calculation for a one-dimensional cylindrical reactor. Two energy-group cross sections are used and the scattering is assumed isotropic. The EDIT module is not executed in this sample.

The reactor model consists of a central core of radius 40 cm surrounded by an annular blanket 30 cm thick followed by a shield 30 cm thick. The core consists of 35 volume percent (v/o) fuel, 40 v/o sodium, and 25 v/o steel. The blanket contains 35 v/o blanket fuel, 40 v/o sodium, and 25 v/o steel. The shield consists of 70 v/o sodium and 30 v/o steel.

The first page of the ONEDANT output lists the entire card-image input "deck" supplied to the ONEDANT code for this sample problem. The code provides this card-image input listing unless the third entry on card 1, the entry NOLIST, is set to unity by the user. Note that numerous "comment cards" have been used in the card-image input using the slash (/) as described in Ch. III.

On page 2 of the problem output are a descriptive summary of the Title Card Control Parameters and the printout of the two title cards provided. This is followed by the message KEY END BLOCK I READ, which indicates that all BLOCK I input has been successfully read and is ready for processing. Next appears the BLOCK I input summary followed by messages that both the BLOCK II and BLOCK III input card-images were successfully read.

On page 3 of the output is a descriptive summary of the BLOCK III card-image input pertaining to cross sections. Included in this summary is a listing of the cross-section types from the card-image library that can be used for edit purposes. These edit cross sections are written to the SNXEDT group-ordered

cross-section interface file for use by the EDIT module, if desired. (See Ch. IV, Table II.) The card-image cross-section library, provided directly in the card-image input, is read and the header cards that were included in the library are printed for the user. For this sample problem cross sections for seven isotopes have been provided. Hollerith names have been provided through the NAMES array in BLOCK III and these are listed under the column labelled Isotope Name. The scattering is specified to be isotropic and this is indicated by the entries "PO" under the column labelled Order. (The label "Order" refers to the Legendre order of expansion for the scattering and, since it is isotropic, only the P_0 Legendre polynomial term appears.)

Page 4 of the output provides the user with a listing of all Nuclide and Material Mixing instructions provided in Block IV of the card-image input. For this problem the nuclides FE (iron), CR (chromium), and NI (nickel) are mixed with atom densities 0.05, 0.016, and 0.01, respectively, to create the Material named STEEL. The mixed-oxide, (U-238, PU-239) O_2 , Material named FUEL is then created using the isotopes PU-239, U-238, and O-16 with atom densities of 0.0051, 0.0206, and 0.0412, respectively. The depleted uranium oxide Material named BLKT and the Material SODIUM are also created as shown in the output. These specifications are provided in the card-image input through the MATLS= input in BLOCK IV. Through the ASSIGN= input in BLOCK IV the four materials STEEL, FUEL, BLKT, and SODIUM are suitably mixed to create the actual macroscopic mixtures assigned to each of the three ZONES in the sample problem: the core zone (named CORE), the blanket zone (named BLANKT), and the outer shield zone (named SHIELD). The CORE consists of the Material FUEL with a volume fraction (density) of 0.35, SODIUM with a volume fraction 0.40, and STEEL with a volume fraction 0.25. The zone BLANKT is identical to the CORE except that the Material FUEL is replaced by the Material BLKT. The SHIELD zone consists only of the Materials SODIUM and STEEL. The subsequent message KEY START MIX CARD XS indicates that the ONEDANT INPUT module is to begin creating the working cross-section files MACRXS and SNXEDT and the standard interface files NDXSRF and ZNATDN as described in Ch. IV. D. Also provided is the data storage requirement in large core memory (LCM) for the cross-section processing. As indicated in the sample problem output, 92 words of LCM are required. The 1,7 following this message refers to the secondary overlay in which the mixing and cross-section processing takes place in this problem (see Fig. 2 in Ch. II). The last three KEY END messages

on the page indicate that the cross-section mixing and processing operation was completed, the BLOCK V SOLVER module input was read (and the SOLINP interface file created), and all INPUT module operations were completed.

Page 5 of the sample problem output begins the printed output provided by the SOLVER module. Pages 5 and 6 present a summary of the input parameters related, to or required by, the SOLVER module as provided (or defaulted). Note that for the input parameters two columns are provided: one labelled RAW INPUT and one labelled AS DEFAULTED. The RAW INPUT column presents the actual input values provided by the user. If no entry is made in the input, a RAW INPUT value of zero is listed. The AS DEFAULTED column lists the values of the input parameters that the SOLVER module actually uses. For example, on page 5 of the output, under the heading CONVERGENCE CONTROLS, the RAW INPUT value for the parameters EPSI is listed as 0. (In the actual card-image input, no entry for EPSI has been provided.) The default value for EPSI (0.0001) is, accordingly, assumed by the SOLVER module and this value is provided under the AS DEFAULTED column.

On page 6 of the output are listed the BLOCK I input parameters that are carried over for use by the SOLVER module. Here, for example, is indicated that the problem is cylindrical geometry (IGEOM= 2), two energy groups (NGROUP= 2), and S_4 quadrature is to be used (ISN= 4), etc.

Page 7 of the output provides a recap of the assignment of materials to zones in terms of the algorithm described in Ch. IV. C under the ASGMOD ARRAY description in BLOCK IV. Following this is a map of the problem geometry showing the coarse-mesh boundary locations, the zone number assigned to each coarse-mesh interval, and other pertinent information. The data storage requirements for the SOLVER module are shown next. Below this is a summary of the discrete-ordinates quadrature quantities used for the calculation. For this problem the values printed are built-in S_4 Gaussian quadrature values. Column headings generally refer to quantities depicted in Fig. 8 in Ch.V. The column labelled LI refers to the ξ -level index. The terms BETA PLUS and BETA MINUS refer, respectively, to the terms $\alpha_{m+\frac{1}{2}}/w_m$ and $\alpha_{m-\frac{1}{2}}/w_m$ in Eq. (28) of Ch. V. (For spherical geometry BETA PLUS and BETA MINUS refer, respectively, to one-half the value of the terms $\beta_{m+\frac{1}{2}}/w_m$ and $\beta_{m-\frac{1}{2}}/w_m$ in Eq. (31) of Ch. V.)

Page 8 lists the material names of materials for which cross-section data exist on the MACRXS interface file being used by the SOLVER module. Next is provided a listing of the ZONE macroscopic cross sections used by the SOLVER module. This print is optional and is controlled by the XSECTP entry in the BLOCK V input. In this sample the full table of ZONE macroscopic cross sections has been requested by setting XSECTP= 2. The PRINCIPAL CROSS SECTIONS are defined as the ZONE macroscopic values of χ (fission fraction), $\nu\bar{\Sigma}_f$, $\bar{\Sigma}_t$, and $\bar{\Sigma}_a$. The scattering matrix terms correspond to the term $\sigma_{s,h\rightarrow g}^n$ in Eq. (19) of Ch. V. The superscript n denotes the Legendre expansion order for the term; the value of n is provided under the column labelled ORDER in the printout. The actual scatter matrix terms for scatter from energy-group h to energy-group g are listed across the page in the sequence

$$\sigma_{s,h\rightarrow g} \quad \sigma_{s,h-1\rightarrow g} \quad \sigma_{s,h-2\rightarrow g} \quad \text{etc.}$$

The entries in the column labelled FIRST in the printout give the value of the energy-group h, namely the first group in the listing which scatters into group g. For downscatter only problems, the value of h is the same as the group number g. For upscatter problems the value of h will not be the same as the value of g. At the bottom of page 8 of the output is geometry and spatial mesh information.

Page 9 of the output provides a summary description of iteration control criteria followed by the iteration monitor print. These items are fully described in Ch. VI. It is noted that for this type of problem, a k_{eff} calculation, the eigenvalue is the value of k_{eff} . For the sample problem, then, $k_{\text{eff}} = 0.993\ 154$ is provided in the monitor print for outer Iteration 4 under the column labelled K-EFF EIGENVALUE.

Page 10 of the output provides a final system edit and balance table print for each energy group and the sum of the groups. The group-dependent quantities are defined and computed as follows:

$$(1) \text{ SOURCE} = \text{total inhomogeneous source} = QG_g =$$

$$\sum_{i=1}^{IT} Q_i v_i + \sum_{\mu_m < 0} w_m |\mu_m| A_{IT+\frac{1}{2}} QR_m + \sum_{\mu_m > 0} w_m \mu_m A_{\frac{1}{2}} QL_m$$

where Q_i is the inhomogeneous distributed source, QL_m is the left boundary (surface) source, QR_m is the right boundary (surface) source, V_i is the "volume" of spatial mesh interval i , $A_{IT+\frac{1}{2}}$ is the surface area at the rightmost boundary of the system, and $A_{\frac{1}{2}}$ is the surface area at the leftmost boundary of the system;

(2) FISSION SOURCE = total fission source to the group $g = FG_g =$

$$\sum_{h=1}^{NGROUP} \sum_{i=1}^{IT} \chi_{g,i}^{(v\sigma_f)_{h,i}} \phi_{h,i} V_i ;$$

(3) IN SCATTER = in scatter source to group g from other groups =

$$SIN_g = \sum_{\substack{h=1 \\ h \neq g}}^{NGROUP} \sum_{i=1}^{IT} (\sigma_{s,h \rightarrow g})_i \phi_{h,i} V_i ;$$

(4) SELF SCATTER = self-scatter (within group scatter) in group $g =$

$$SS_g = \sum_{i=1}^{IT} (\sigma_{s,g \rightarrow g}^o)_i \phi_{g,i} V_i ;$$

(5) ABSORPTION = absorption in group $g =$

$$ABG_g = \sum_{i=1}^{IT} (\sigma'_{a,g})_i \phi_{g,i} V_i ,$$

where $\sigma'_{a,g}$ is the absorption cross section for group g plus any buckling "absorption" plus any "time absorption" (α/v_g);

(6) OUT SCATTER = out scatter from group g to all other groups =

$$\text{SOUT}_g = \sum_{i=1}^{IT} (\sigma'_{t,g})_i \phi_{g,i} V_i ,$$

where $\sigma'_{t,g}$ is the total cross section for group g plus any buckling "absorption" plus any "time absorption" (α/v_g);

(7) RIGHT LEAKAGE = net flow out of system right boundary =

$$\text{RL}_g = \sum_{\mu_m > 0} w_m \mu_m A_{IT+\frac{1}{2}} \psi_{m,IT+\frac{1}{2}} - \sum_{\mu_m < 0} w_m |\mu_m| A_{IT+\frac{1}{2}} \psi_{m,IT+\frac{1}{2}} ;$$

(8) NET LEAKAGE = Net flow from system (both boundaries) =

$$\text{NL}_g = \text{RL}_g + \sum_{\mu_m < 0} w_m |\mu_m| A_{\frac{1}{2}} \psi_{m,\frac{1}{2}} - \sum_{\mu_m > 0} w_m \mu_m A_{\frac{1}{2}} \psi_{m,\frac{1}{2}} ;$$

and

(9) PARTICLE BALANCE =

$$\text{BAL}_g = 1 - \frac{\text{NL}_g + \text{ABG}_g + \text{SOUT}_g}{\text{QG}_g + \text{FG}_g + \text{SIN}_g} .$$

Page 11 of the sample problem output provides two optional pointwise quantity printouts. The isotropic flux print is provided when the input parameter FLUXP is BLOCK V is set to a value of 1 or 2 (a value of unity is used in this sample problem). The flux values printed are the mesh-interval average fluxes, commonly referred to as the cell-centered flux values. The fission source rate print is provided when the input parameter FISSRP in BLOCK V is set to unity (as in this sample problem). The fission source rate for energy-group g and mesh point i is simply the quantity $(v\sigma_f)_{g,i} \phi_{g,i}$, having units of particles per unit time and volume.

Page 12 of the output, the final page, provides the RUN HIGHLIGHTS for the sample problem execution. This is followed by a storage and timing history of the run by primary and secondary overlay as shown in Figs. 2 and 3 of Ch. II.

It is to be noted that no EDIT module output appears in the output of this sample problem. The reason for this is that no EDIT module input (BLOCK VI of the card-image input) is provided in the input "deck" and no EDITIT binary interface file (containing previously created EDIT module input) was in existence at the time of the sample problem execution.

GENERALIZED INPUT MODULE RUN ON 11/05/81 WITH VERSION 11-02-B1

...LISTING OF CARDS IN THE INPUT STREAM...

```

1.      2      O      O
2.  SAMPLE PROBLEM 1 FOR USER'S MANUAL
3.  STANDARD K CALCULATION, ALL INPUT BY MEANS OF CARD-IMAGES
4.  / GEOMETRY - CYLINDRICAL
5.  / CROSS SECTIONS - 2 GROUP, ISOTROPIC SCATTER
6.  / ISOTOPE DATA ON CARDS, LOS ALAMOS (DTF) FORMAT
7.  / MIXING - ISOTOPES MIXED TO MAKE MATERIALS NAMED STEEL,
8.  / FUEL, BLKT, AND SODIUM
9.  / MATERIALS ASSIGNED TO MAKE ZONES NAMED CORE,
10. / BLANKT, AND SHIELD.
11. / SOLVER - CARD INPUT SUPPLIED
12. / EDITS - NONE
13. /
14. /
15. / * * * * BLOCK I * * * *
16. IGEOM=2, NGROUP=2, ISN=4 NISO=7 MT=4 NZONE=3 IM=3 IT=50 T
17. /
18. /
19. / * * * * BLOCK II (GEOMETRY) * * * *
20. XMESH=0 0.40,70,100 XINTS= 20, 2R15
21. ZONES= 1 2 3 T
22. /
23. /
24. / * * * * BLOCK III (CROSS SECTIONS) * * * *
25. LIB= ODNINP
26. MAXDRO=0 IHM=6 IHT=4 IHS=5 IFIDO=0 ITITL=1
27. NAMES= "0-16" "NA-23" FE CR NI "PU-239" "U-238"
28. EDNAME= FISS
29. /
30. / * * * * SINCE LIB=ODNINP, THE CROSS SECTION LIBRARY IN CARD-IMAGES
31. / WILL BEGIN IMMEDIATELY FOLLOWING THE BLOCK III TERMINAL "T".
32. / NOTE THAT A TITLE CARD PRECEDES EACH CROSS SECTION
33. / BLOCK (SINCE ITITL=1). * * * *
34. /
35. T
36. OXYGEN-16 (0-16) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
37. 0.000 0.010 0.000 2.000 1.600 0.000 016/1
38. 0.000 0.000 0.000 0.000 3.600 0.390 016/2
39. SODIUM (NA-23) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
40. 0.000 0.002 0.000 1.900 1.500 0.000 NA23/1
41. 0.000 0.005 0.000 4.000 3.995 0.398 NA23/2
42. IRON (FE) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
43. 0.000 0.008 0.000 2.100 1.700 0.000 FE/1
44. 0.000 0.010 0.000 4.500 4.490 0.392 FE/2
45. CHROMIUM (CR) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
46. 0.000 0.013 0.000 2.450 2.150 0.000 CR/1
47. 0.000 0.020 0.000 5.000 4.980 0.287 CR/2
48. NICKEL (NI) SAMPLE 2 GRDUP LMFBR CROSS SECTIONS
49. 0.000 0.080 0.000 2.400 2.000 0.000 NI/1
50. 0.000 0.030 0.000 8.000 7.970 0.320 NI/2
51. PLUTONIUM (PU-239) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
52. 1.900 1.950 6.270 4.800 2.000 0.000 PU239/1
53. 1.600 2.500 4.800 12.000 9.500 0.850 PU239/2
54. URANIUM (U-238) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
55. 0.300 0.400 0.900 4.700 3.000 0.000 U238/1
56. 0.000 0.500 0.000 13.000 12.500 1.300 U238/2
57. /
58. / * * * * END OF CROSS SECTION DATA * * * *
59. / * * * * NOTE THAT THERE IS NO TERMINAL "T" SINCE THE CROSS SECTIONS ARE
60. / IN LOS ALAMOS (DTF) FORMAT (IFIDO=0) * * * *
61. /
62. /
63. / * * * * BLOCK IV (MIXING) * * * *
64. MATLS= STEEL .05, CR .016, NI 0.01;
65. FUEL "PU-239" .0051, "U-238" .0155 "0-16" .0412;
66. BLKT "U-238" 0.0206, "D-16" .0412;
67. SODIUM "NA-23" .025
68. ASSIGN= CORE FUEL .35, SODIUM .4, STEEL .25;
69. BLANKT BLKT .35, SODIUM .4, STEEL .25;
70. SHIELD SODIUM .7, STEEL .3 T
71. /
72. /
73. / * * * * BLOCK V (SOLVER) * * * *
74. IEVT=1 ISCT=0 IBR=0
75. NORM=1 FLUXP=1 XSECTP=2 FISSRP=1
76. CHI=0.6,0.4; 0.7, 0.3 T
77. /
78. /
79. / * * * * BLOCK VI (EDITS) * * * *
80. / NO INFORMATION SUPPLIED

```

```

.....
.
.
.
          CASE TITLE
.....
*KEY START CASE INPUT *
.....
.
.
.
          2 NHEAD  NUMBER OF TITLE CARDS TO FOLLOW
          0 NOTTY  0/1 NO/YES  SUPPRESS ON-LINE TERMINAL  OUTPUT
          0 NOLIST 0/1 NO/YES  SUPPRESS INPUT LISTING
.....
*
*
* SAMPLE PROBLEM 1 FOR USER'S MANUAL
* STANDARD K CALCULATION. ALL INPUT BY MEANS OF CARD IMAGES
*
*
.....
*KEY END  BLOCK I READ*
.....

```

```

.....
.
.
.
          ...BLOCK I - CONTROLS AND DIMENSIONS...
.....
.
.
.
          ...DIMENSIONS (ARRAY NAME = DIMENS)...
.....
          2 IGEOM  1/2/3  PLANE/CYLINDER/SPHERE
          2 NGROUP NUMBER OF ENERGY GROUPS
          4 ISN    ANGULAR QUADRATURE ORDER
          7 NISD   NUMBER OF INPUT ISOTOPES (FROM ISOTXS, GRUPXS, DR CARDS)
          4 MT    NUMBER OF PERMANENT MATERIALS
          3 NZONE NUMBER OF ZONES
          3 IM    NUMBER OF COARSE MESH X INTERVALS
          50 IT   NUMBER OF FINE MESH X INTERVALS
.....
          ...STORAGE...
          MAXLCM= 50000
.....
*KEY END  BLOCK II READ-GEOM*
.....
*KEY END  BLOCK III READ-XS *
.....

```

.....BLOCK III - CROSS SECTION LIBRARY.....

.....LIBRARY SOURCE...
LIB=ODNINP

.....CARD LIBRARY PARAMETERS (ARRAY NAME = CARDS).....

0 MAXORD MAXIMUM LEGENDRE ORDER TO BE FOUND IN INPUT CROSS SECTIONS
6 IHM LAST POSITION IN CROSS SECTION TABLE
4 IHT POSITION OF TOTAL CROSS SECTION
5 IHS POSITION OF SELF SCATTER CROSS SECTION
0 IFIDO O/1/2 - DTF/FIXED FIDO/FREE FIDO LIBRARY
1 ITITL O/1 - NO/YES THERE IS A TITLE CARD BEFORE EACH TABLE
0 I2LP1 O/1 - NO/YES LIBRARY HIGHER ORDER SCATTERING CONTAINS 2L+1 FACTOR
0 SAVBXS O/1 - NO/YES SAVE BINARY XSLIB FILE (FILENAME=BXSLIB)
1 KWIKRD O/1 - FULL FIDO READ/QUICK FIDO READ (DEFAULT=QUICK)

.....ENERGY STRUCTURE.....

GROUP	VEL	LOWER BOUND	UPPER BOUND	GROUP	VEL	LOWER BOUND	UPPER BOUND
1	0.	0.	0.	2	0.	0.	0.

LAST NEUTRON GROUP(LNG) IS NUMBER 2

.....EDIT POSITION NAMES.....

POSITION	EDNAME	CARD POSITION
1	CHI	
2	NUSIGF	3
3	TOTAL	4
4	ABS	2
5	FISS	1

*KEY START CARD LIBE READ *

.....HEADER CARDS FROM THE CARD LIBRARY.....

ISOTOPE NUMBER	ISOTOPE NAME	OROR	HEADER CARD
1.	O-16	PO -	OXYGEN-16 (O-16) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
2.	NA-23	PO -	SODIUM (NA-23) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
3.	FE	PO -	IRON (FE) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
4.	CR	PO -	CHROMIUM (CR) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
5.	NI	PO -	NICKEL (NI) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
6.	PU-239	PO -	PLUTONIUM (PU-239) SAMPLE 2 GROUP LMFBR CROSS SECTIONS
7.	U-238	PO -	URANIUM (U-238) SAMPLE 2 GROUP LMFBR CROSS SECTIONS

*KEY END CARD LIBE READ *

KEY END BLOCK IV READ MATS


```

.....
*
*                                ... MIXING INSTRUCTIONS ...
*
.....

```

```

*
*   MIX      COMP      DENSITY      COMP      DENSITY ETC.
*   ----      ----      -
*
*   MATLS
*   -----
*
*   1. STEEL  FE       5.00000E-02, CR   1.60000E-02, NI   1.00000E-02,
*   2. FUEL   PU-239  5.10000E-03, U-238  1.55000E-02, O-16  4.12000E-02,
*   3. BLKT   U-238  2.06000E-02, O-16  4.12000E-02,
*   4. SODIUM NA-23  2.50000E-02,

```

```

*
*   ASSIGN
*   -----
*
*   1. CDPE   FUEL    3.50000E-01, SDDIUM 4.00000E-01, STEEL  2.50000E-01,
*   2. BLANKT BLKT    3.50000E-01, SDDIUM 4.00000E-01, STEEL  2.50000E-01,
*   3. SHIELD SDDIUM  7.00000E-01, STEEL  3.00000E-01,

```

```

*KEY START MIX CARD XS *
.....

```

```

92 WORDS LCM REQUIRED 1,7
.....

```

```

*KEY END MIX CARD XS *
.....

```

```

*KEY END BLOCK V READ-SOLVR*
.....

```

```

*KEY END INPUT MODULE*
.....
*
*
*
.....

```

5.

```

.....
THIS ONEDANT PROBLEM RUN ON 11/05/81 WITH VERSION 11-02-81
.....

```

```

.....
*KEY START SOLVER EXECUTION *
.....

```

```

.....
...TITLE...

```

```

SAMPLE PROBLEM 1 FOR USER'S MANUAL
STANDARD K CALCULATION, ALL INPUT BY MEANS OF CARD-IMAGES
.....

```

```

.....
...BLOCK V -- SOLVER INPUT...
.....

```

```

RAW    AS
INPUT  DEFAULTED
-----

```

```

.....
...REQUIRED INPUT (ARRAY NAME = SOLIN)...

```

RAW INPUT	AS DEFAULTED		
1	1	IEVT	-1/0/1/2/3/4 - TYPE OF CALCULATION -1 INHOMOGENEOUS SOURCE WITH FISSION AND/OR UPSCATTER 0 INHOMOGENEOUS SOURCE ALONE 1 K-EFFECTIVE 2 ALPHA OR TIME ABSORPTION SEARCH 3 CONCENTRATION SEARCH 4 DELTA (I.E. DIMENSION) SEARCH
0	0	ISCT	LEGENDRE ORDER OF SCATTERING
0	0	ITH	0/1 - DIRECT/ADJOINT - MODE OF CALCULATION (DEFAULT=DIRECT)
0	1	IBL	0/1/2/3 - LEFT BOUNDARY CONDITION VACUUM/REFLECTIVE/PERIODIC/WHITE
0	0	IBR	0/1/2/3 - RIGHT BOUNDARY CONDITION VACUUM/REFLECTIVE/PERIODIC/WHITE

```

.....
...CONVERGENCE CONTROLS (ARRAY NAME = ITER)...

```

RAW INPUT	AS DEFAULTED		
0.	1.000E-04	EPSI	INNER ITERATION CONVERGENCE CRITERION (DEFAULT=0.0001)
0.	1.000E-04	EPSO	OUTER ITERATION CONVERGENCE CRITERION (DEFAULT=EPSI)
0	1	IIIL	MAXIMUM NUMBER OF INNER ITERATIONS PER GROUP UNTIL (1-LAMBDA).LT.3*EPSO (DEFAULT CALCULATED)
0	8	IIIM	MAXIMUM NUMBER OF INNER ITERATIONS PER GROUP AFTER (1-LAMBDA).LT.3*EPSO (DEFAULT CALCULATED)
0	20	OITM	MAXIMUM NUMBER OF OUTER ITERATIONS (DEFAULT=20)
0	0	ITLIM	ITERATION TIME LIMIT (SECONDS)

```

.....

```

```
.....
..BLOCK V -- SOLVER INPUT (CONTINUED)..
.....
```

```
RAW      AS
INPUT  DEFAULTED
-----
```

```
.....MISCELLANEDUS PARAMETERS (ARRAY NAME = MISC).....
```

```
0.      0.      BHGT  BUCKLING HEIGHT
0.      0.      BWTH  BUCKLING WIDTH (PLANE GEOM ONLY)
1.000E+00 1.000E+00 NORM  NORMALIZATION FACTOR

0      0      INFLUX 0/1 NO/YES - READ INPUT FLUX FROM FILE RTFLUX (ATFLUX FOR ADJOINT)
0      0      INSORS 0/1 NO/YES - READ INPUT SOURCE FROM FILE FIXSRC
0      0      I2ANG 0/1 NO/YES - DO 2 ANGLE CALCULATION
C      1      IQAD  -3/1/2/3 - SOURCE OF QUADRATURE CONSTANTS (DEFAULT=1)
                           FILE SNCONS/BUILT IN P-N/BUILT IN DP-N/CARDS
```

```
.....OUTPUT CONTROLS (ARRAY NAME = SOLOUT).....
```

```
1 FLUXP 0/1/2 NONE/ISOTROPIC/ALL MOMENTS - FLUX PRINT
2 XSECT 0/1/2 NONE/PRINCIPAL/ALL - MACROSCOPIC CROSS SECTION PRINT
1 FISSRP 0/1 NO/YES - PRINT FINAL FISSION SOURCE RATE
0 SOURCP 0/1/2/3 NO/AS READ/NORMALIZED/BOTH - PRINT INHOMOGENEDUS SOURCE
0 GEOMP 0/1 NO/YES - PRINT FINE MESH GEOMETRY
0 ANGP 0/1 NO/YES - PRINT ANGULAR FLUXES
0 RAFLUX 0/1 NO/YES - WRITE ANGULAR FLUXES TO FILE RAFLUX (AAFLUX IF ADJOINT)
0 BALP 0/1 NO/YES - PRINT COARSE MESH BALANCES (CURRENTLY UNIMPLEMENTED)
```

```
.....PARAMETERS INFERRED FROM INPUT ARRAYS.....
```

```
2 INCHI 0/1/2 NONE/ONE CHI/ZONEWISE CHI
0 IDEN 0/1 NO/YES - USE DENSITY FACTORS
0 IOAN SOURCE ANISOTROPY
0 ISORSE NUMBER OF SOURCE VECTORS INPUT
0 ISORSX NUMBER OF SOURCE VECTORS INPUT
0 ISORSF NUMBER OF SOURCE VECTORS INPUT
0 IQL -1/0/1 ISDTROPIC/NONE/ALL ANGLES - LEFT BOUNDARY SOURCE
0 IQR -1/0/1 ISDTROPIC/NONE/ALL ANGLES - RIGHT BOUNDARY SOURCE
0 ISBEDD 0/N NO/YES - THERE ARE ALBEDOES
```

```
.....PARAMETERS FROM BLOCK I.....
```

```
2 IGEOM 1/2/3 PLANE/CYLINDER/SPHERE
2 NGROUP NUMBER OF ENERGY GROUPS
4 ISN ANGULAR QUADRATURE ORDER
4 MT NUMBER OF PERMANENT MATERIALS
3 NZONE NUMBER OF ZONES
3 IM NUMBER OF COARSE MESH X INTERVALS
50 IT NUMBER OF FINE MESH X INTERVALS
```

...MATERIAL ASSIGNMENTS TO ZONES...

KEY START MATLS TO ZONES

ZONE CROSS SECTION = SUM OVER MATLS IN THE ZONE OF (MATL CROSS SECTION)*(CO + C1*CMOD)
WITH
CMOD = 0.

ENTRY	ZONE NO.	ZONE NAME	MATERIAL NO.	MATERIAL NAME	CO	C1
1	1	CORE	2	FUEL	3.500000E-01	0.
2	1	CORE	4	SODIUM	4.000000E-01	0.
3	1	CORE	1	STEEL	2.500000E-01	0.
4	2	BLANKT	3	BLKT	3.500000E-01	0.
5	2	BLANKT	4	SODIUM	4.000000E-01	0.
6	2	BLANKT	1	STEEL	2.500000E-01	0.
7	3	SHIELD	4	SDDIUM	7.000000E-01	0.
8	3	SHIELD	1	STEEL	3.000000E-01	0.

...ZONE ASSIGNMENT MAP...

KEY START MAP

```

      111111111111111111111111
      1      *      *      0
      1      1 *      2 *      3 0
      1      *      *      0
      R      0.      40.      70.      100.
      0000      0000      0000      0000

```

```

XINTS      20      15      15
COARSE MESH 1      2      3

```

...COMMENTS FROM ONEDANT ON STORAGE REQUIREMENTS...

KEY START STORAGE NOTE

STORAGE	REQUIRED	ALLOWED
SMALL CORE	944	25400
LARGE CORE	671	50000

...ANGULAR QUADRATURE DATA FOR ONEDANT...

XI-LEVEL	STARTING COSINE
1	-5.083741E-01
2	-9.404323E-01

MU-LEVEL	LEVEL WEIGHT	LEVEL COSINE
1	1.739274E-01	-8.611363E-01
2	3.260726E-01	-3.399810E-01
3	3.260726E-01	3.399810E-01
4	1.739274E-01	8.611363E-01

M	REFL	M	LI	POINT WEIGHT	MU COSINE	WGT*MU	BETA PLUS	BETA MINUS	XI	PHI(DEG)
1	4	1	1.630363E-01	-3.399810E-01	-5.542925E-02	3.399810E-01	0.	86113631	131.9715	
2	6	2	1.739274E-01	-8.611363E-01	-1.497752E-01	8.611363E-01	0.	33998104	156.3027	
3	5	2	1.630363E-01	-3.399810E-01	-5.542925E-02	1.258643E+00	9.186619E-01	33998104	111.1933	
4	1	1	1.630363E-01	3.399810E-01	5.542925E-02	0.	3.399810E-01	86113631	48.0285	
5	3	2	1.630363E-01	3.399810E-01	5.542925E-02	9.186619E-01	1.258643E+00	33998104	68.8067	
6	2	2	1.739274E-01	8.611363E-01	1.497752E-01	0.	8.611363E-01	33998104	23.6973	

...FISSION GUESS-FLUX ZEROED IN ALL GROUPS...

...CROSS SECTION RELATED DATA FROM FILE MACRXS 154032110581 VERSION 1...

...MATERIALS AVAILABLE FROM FILE MACRXS...

1 STEEL 2 FUEL 3 BLKT 4 SODIUM

...CROSS SECTIONS FOR LEGENDRE ORDERS UP TO PO...

KEY START MAC CROSS SECTIONS

GROUP 1

...PRINCIPAL CROSS SECTIONS...

ZONE NO.	NAME	CHI	NU*FISSION	TOTAL	ABSORPTIDN
1	CORE	6.0000E-01	1.6074E-02	1.2396E-01	6.1669E-03
2	BLANKT	7.0000E-01	6.4890E-03	1.2378E-01	3.4002E-03
3	SHIELD	7.0000E-01	0.	8.3710E-02	4.5740E-04

...SCATTERING MATRICES...

ZJNE	ORDER	FIRST	CROSS SECTIONS
1	0	1	9.2767E-02
2	0	1	9.4552E-02
3	0	1	6.8070E-02

GROUP 2

...PRINCIPAL CROSS SECTIONS...

ZONE NO.	NAME	CHI	NU*FISSION	TOTAL	ABSORPTION
1	CORE	4.0000E-01	8.5680E-03	2.8011E-01	7.5050E-03
2	BLANKT	3.0000E-01	0.	2.8189E-01	3.9350E-03
3	SHIELD	3.0000E-01	0.	1.8550E-01	4.2350E-04

...SCATTERING MATRICES...

ZONE	ORDER	FIRST	CROSS SECTIONS
1	0	2	2.7260E-01 2.5022E-02
2	0	2	2.7796E-01 2.5825E-02
3	0	2	1.8508E-01 1.5183E-02

...GEOMETRY INFORMATION AS EDITED BY ONEDANT...

...COARSE MESH GEOMETRY...

	NO. OF INTERVALS	WIDTH	FINE MESH SIZE	LEFT BOUNDARY
1	20	4.00000000E+01	2.00000000E+00	0.
2	15	3.00000000E+01	2.00000000E+00	4.00000000E+01
3	15	3.00000000E+01	2.00000000E+00	7.00000000E+01
				1.00000000E+02

... ITERATION CONTROLS AND CRITERIA ...

*** ITERATION CRITERIA ***

TRANSPORT INNERS

CRITERION	QUANTITY TO TEST	VALUE	ACTION TAKEN IF VALUE EXCEEDED
IITL	INNER ITERATION COUNT UNTIL (1.-LAMBDA).LT.3*EPSO	1	TERMINATES INNERS
IITM	INNER ITERATION COUNT AFTER (1.-LAMBDA).LT.3*EPSO	8	TERMINATES INNERS
EPSI	FRACTIONAL PTWISE FLUX CHANGE PER INNER	1.00E-04	DOES ANOTHER INNER

DIFFUSION SUB-OUTERS

CRITERION	QUANTITY TO TEST	VALUE	ACTION TAKEN IF VALUE EXCEEDED
QITMD	SUB-OUTER ITERATION COUNT	100	TERMINATES SUB-OUTERS
EPSO	DIFFUSION LAMBDA-1.0	1.00E-04	DOES ANOTHER SUB-OUTER
EPSX	FRACTIONAL LAMBDA CHANGE PER SUB-OUTER	2.98E-04	DOES ANOTHER SUB-OUTER
EPSA	FRACTIONAL PTWISE FLUX CHANGE PER SUB-OUTER	0.95*EPSX	DOES ANOTHER SUB-OUTER

FINAL CONVERGENCE CRITERIA

CRITERION	QUANTITY TO TEST	VALUE	ACTION TAKEN IF VALUE EXCEEDED
OITM	OUTER ITERATION COUNT	20	QUITS WITH ERROR MESSAGE
EPSO	LAMBDA-1.0	1.00E-04	DOES ANOTHER OUTER
EPSX	MAX FRACTIONAL PTWISE FLUX CHANGE	2.98E-04	DOES ANOTHER OUTER

... FLUX AND EIGENVALUE CONVERGENCE AS MONITORED BY ONEDANT ...

*KEY START ITERATION MONITOR *

TIME (SEC)	OUTER NO.	INNER INNERS	DIFFUSION SUB-OUTERS	K-EFF EIGENVALUE	LAMBDA-1	MAX PTWISE FLUX CHANGE	INNER CONVERGED
.06	0						
.12	1	0	8	9.87880E-01	2.71041E-05	6.71213E-04	YES
.14	2	2	3	9.93480E-01	4.57919E-03	8.75653E-02	**NO**
.16	3	2	2	9.93410E-01	-6.25558E-05	1.63735E-02	**NO**

-- INNER ITERATION SUMMARY FOR OUTER ITERATION NO. 4 --

GROUP	ITER	PER GRDUP	MAX FLUX CHANGE	AT MESH
1	3		.57E-04	47
2	2		.87E-04	50

\$\$\$\$\$ ALL CONVERGENCE CRITERIA SATISFIED \$\$\$\$\$

TIME (SEC)	OUTER NO.	INNER INNERS	DIFFUSION SUB-OUTERS	K-EFF EIGENVALUE	LAMBDA-1	MAX PTWISE FLUX CHANGE	INNER CONVERGED
.27	4	5	1	9.93402E-01	-2.96675E-06	1.47891E-04	YES

PARTICLE BALANCE = -1.61963E-06 TOTAL INNERS ALL OUTERS = 9

```

.....
..
.
...GROUP EDIT AND BALANCES UPON CONVERGENCE...
.
.....
.
...TITLE--- SAMPLE PROBLEM 1 FOR USER'S MANUAL
.
.
.
...SYSTEM BALANCE TABLES...
.
.....
*KEY START BALANCE TABLES*
.....

```

GP	SOURCE	FISSION SOURCE	IN SCATTER	SELF SCATTER	OUT SCATTER
1	0.	6.0245109E-01	-3.5527137E-15	1.8227508E+00	4.9157276E-01
2	0.	3.9754891E-01	4.9157276E-01	3.9595671E+01	-4.5474735E-13
TOTAL	0.	1.0000000E+00	4.9157276E-01	4.1418422E+01	4.9157276E-01

GP	ABSORPTION	PARTICLE BALANCE	RIGHT LEAKAGE	NET LEAKAGE
1	1.0871255E-01	-1.4088478E-07	2.1659568E-03	2.1659568E-03
2	8.3207468E-01	-2.6216060E-06	5.7049228E-02	5.7049228E-02
TOTAL	9.4078723E-01	-1.6196345E-06	5.9215185E-02	5.9215185E-02

```

.....
.
.
EXECUTION TIME          .28 SECONDS
.
.
.

```

GROUP FLUXES AND FISSION SOURCE RATES BY SPATIAL MESH...

KEY START FLUXES:

GROUP 1

ISOTROPIC FLUX COMPONENT

ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT				
1	1	4	7570E-03	1	11	3.8007E-03	2	21	1.2723E-03	2	31	1.5403E-04	3	41	3.0295E-05
1	2	4	7424E-03	1	12	3.6190E-03	2	22	9.9167E-04	2	32	1.2685E-04	3	42	2.6351E-05
1	3	4	7051E-03	1	13	3.4237E-03	2	23	7.8766E-04	2	33	1.0420E-04	3	43	2.2860E-05
1	4	4	6499E-03	1	14	3.2151E-03	2	24	6.3301E-04	2	34	8.5762E-05	3	44	1.9760E-05
1	5	4	5771E-03	1	15	2.9933E-03	2	25	5.1254E-04	2	35	7.0604E-05	3	45	1.6993E-05
1	6	4	4874E-03	1	16	2.7579E-03	2	26	4.1705E-04	3	36	5.9743E-05	3	46	1.4505E-05
1	7	4	3813E-03	1	17	2.5077E-03	2	27	3.4048E-04	3	37	5.2212E-05	3	47	1.2346E-05
1	8	4	2590E-03	1	18	2.2395E-03	2	28	2.7863E-04	3	38	4.5621E-05	3	48	1.0165E-05
1	9	4	1212E-03	1	19	1.9466E-03	2	29	2.2841E-04	3	39	3.9842E-05	3	49	8.2077E-06
1	10	3	9682E-03	1	20	1.6146E-03	2	30	1.8748E-04	3	40	3.4765E-05	3	50	6.3103E-06

FISSION SOURCE RATE

ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT				
1	1	7	6466E-05	1	11	6.1094E-05	2	21	8.2562E-06	2	31	9.9952E-07	3	41	0.
1	2	7	6231E-05	1	12	5.8174E-05	2	22	6.4350E-06	2	32	8.2165E-07	3	42	0.
1	3	7	5620E-05	1	13	5.5035E-05	2	23	5.1112E-06	2	33	6.4615E-07	3	43	0.
1	4	7	4745E-05	1	14	5.1682E-05	2	24	4.1076E-06	2	34	5.5651E-07	3	44	0.
1	5	7	3575E-05	1	15	4.8116E-05	2	25	3.3259E-06	2	35	4.5815E-07	3	45	0.
1	6	7	2133E-05	1	16	4.4332E-05	2	26	2.7062E-06	3	36	0.	3	46	0.
1	7	7	1042E-05	1	17	4.0309E-05	2	27	2.2094E-06	3	37	0.	3	47	0.
1	8	6	8461E-05	1	18	3.5999E-05	2	28	1.8081E-06	3	38	0.	3	48	0.
1	9	6	6246E-05	1	19	3.1291E-05	2	29	1.4821E-06	3	39	0.	3	49	0.
1	10	6	3787E-05	1	20	2.5954E-05	2	30	1.2165E-06	3	40	0.	3	50	0.

GROUP 2

ISOTROPIC FLUX COMPONENT

ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT				
1	1	2	4258E-02	1	11	1.9697E-02	2	21	9.8549E-03	2	31	3.3104E-03	3	41	1.2578E-03
1	2	2	4171E-02	1	12	1.8855E-02	2	22	8.9004E-03	2	32	2.9565E-03	3	42	1.1331E-03
1	3	2	2990E-02	1	13	1.7960E-02	2	23	8.0172E-03	2	33	2.6410E-03	3	43	1.0115E-03
1	4	2	2726E-02	1	14	1.7021E-02	2	24	7.2069E-03	2	34	2.3607E-03	3	44	8.9296E-04
1	5	2	2378E-02	1	15	1.6043E-02	2	25	6.4678E-03	2	35	2.1138E-03	3	45	7.7713E-04
1	6	2	2249E-02	1	16	1.5033E-02	2	26	5.7966E-03	3	36	1.9285E-03	3	46	6.6379E-04
1	7	2	2441E-02	1	17	1.4001E-02	2	27	5.1893E-03	3	37	1.7890E-03	3	47	5.5245E-04
1	8	2	1859E-02	1	18	1.2955E-02	2	28	4.6415E-03	3	38	1.6515E-03	3	48	4.4221E-04
1	9	2	1204E-02	1	19	1.1907E-02	2	29	4.1487E-03	3	39	1.5170E-03	3	49	3.3116E-04
1	10	2	0482E-02	1	20	1.0867E-02	2	30	3.7065E-03	3	40	1.3857E-03	3	50	2.1499E-04

FISSION SOURCE RATE

ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT				
1	1	2	0784E-04	1	11	1.6876E-04	2	21	0.	2	31	0.	3	41	0.
1	2	2	0709E-04	1	12	1.6155E-04	2	22	0.	2	32	0.	3	42	0.
1	3	2	0555E-04	1	13	1.5389E-04	2	23	0.	2	33	0.	3	43	0.
1	4	2	0328E-04	1	14	1.4583E-04	2	24	0.	2	34	0.	3	44	0.
1	5	2	0030E-04	1	15	1.3745E-04	2	25	0.	2	35	0.	3	45	0.
1	6	1	9663E-04	1	16	1.2881E-04	2	26	0.	3	36	0.	3	46	0.
1	7	1	9228E-04	1	17	1.1996E-04	2	27	0.	3	37	0.	3	47	0.
1	8	1	8728E-04	1	18	1.1100E-04	2	28	0.	3	38	0.	3	48	0.
1	9	1	8167E-04	1	19	1.0202E-04	2	29	0.	3	39	0.	3	49	0.
1	10	1	7549E-04	1	20	9.3109E-05	2	30	0.	3	40	0.	3	50	0.

GROUP SUM

ISOTROPIC FLUX COMPONENT

ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT				
1	1	2	9015E-02	1	11	2.3498E-02	2	21	1.1128E-02	2	31	3.4644E-03	3	41	1.2881E-03
1	2	2	8913E-02	1	12	2.2474E-02	2	22	9.8921E-03	2	32	3.0832E-03	3	42	1.1595E-03
1	3	2	8696E-02	1	13	2.1384E-02	2	23	8.8048E-03	2	33	2.7452E-03	3	43	1.0344E-03
1	4	2	8315E-02	1	14	2.0276E-02	2	24	7.8359E-03	2	34	2.4464E-03	3	44	9.1272E-04
1	5	2	7955E-02	1	15	1.9036E-02	2	25	6.9803E-03	2	35	2.1844E-03	3	45	7.9413E-04
1	6	2	7436E-02	1	16	1.7791E-02	2	26	6.2136E-03	3	36	1.9883E-03	3	46	6.7829E-04
1	7	2	6822E-02	1	17	1.6509E-02	2	27	5.5298E-03	3	37	1.8412E-03	3	47	5.6470E-04
1	8	2	6117E-02	1	18	1.5195E-02	2	28	4.9201E-03	3	38	1.6971E-03	3	48	4.5238E-04
1	9	2	5325E-02	1	19	1.3853E-02	2	29	4.3771E-03	3	39	1.5568E-03	3	49	3.9937E-04
1	10	2	4450E-02	1	20	1.2482E-02	2	30	3.8939E-03	3	40	1.4205E-03	3	50	2.2130E-04

FISSION SOURCE RATE

ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT	ZONE	PT				
1	1	2	8431E-04	1	11	2.2986E-04	2	21	8.2562E-06	2	31	9.9952E-07	3	41	0.
1	2	2	8332E-04	1	12	2.1972E-04	2	22	6.4350E-06	2	32	8.2165E-07	3	42	0.
1	3	2	8118E-04	1	13	2.0892E-04	2	23	5.1112E-06	2	33	6.7615E-07	3	43	0.
1	4	2	7803E-04	1	14	1.9752E-04	2	24	4.1076E-06	2	34	5.5651E-07	3	44	0.
1	5	2	7398E-04	1	15	1.8575E-04	2	25	3.3259E-06	2	35	4.5815E-07	3	45	0.
1	6	2	6876E-04	1	16	1.7314E-04	2	26	2.7062E-06	3	36	0.	3	46	0.
1	7	2	6270E-04	1	17	1.6027E-04	2	27	2.2094E-06	3	37	0.	3	47	0.
1	8	2	5574E-04	1	18	1.4700E-04	2	28	1.8081E-06	3	38	0.	3	48	0.
1	9	2	4792E-04	1	19	1.3331E-04	2	29	1.4821E-06	3	39	0.	3	49	0.
1	10	2	3927E-04	1	20	1.1906E-04	2	30	1.2165E-06	3	40	0.	3	50	0.

INTERFACE FILE SMCNS WRITTEN...

INTERFACE FILE RTFLUX WRITTEN...

II. Sample Problem 2 : Edit-only Run.

Sample Problem 2 is an edit calculation for the problem specified in the first sample problem. The edits are performed using the scalar fluxes produced during the execution of the SOLVER module in Sample Problem 1.

Sample Problem 2 illustrates the way in which the modular construction of ONEDANT can be used to execute the EDIT module independently and separately from a previous SOLVER module execution. The card-image input is shown on the first page of the printed output provided by ONEDANT for Sample Problem 2. Only BLOCK I and BLOCK VI input data are present in the card-image input. The geometry, cross-section, material mixing, and SOLVER portions of the code are thus not executed. Instead, the binary interface files GEODST (geometry), NDXSRF and ZNATDN (mixing), SNXEDT (cross sections for edits), and RTFLUX (scalar fluxes), which were created during a prior execution of Sample Problem 1, were saved and made available to ONEDANT at the time of execution of the second sample problem. This procedure is described in Ch. VIII.

It should be noted that the execution of the EDIT module could have been included in Sample Problem 1 simply by including the BLOCK VI input in the input "deck" for that problem.

Page 1 of the output displays the card-image input for this sample problem. Note the use of comment cards as denoted by the slash (/) as the first entry on each such card-image. Also provided on page 1 are a summary of the Title Card Control Parameters and the printout of the two title cards. Next appears the BLOCK I input summary. Following this appears the message "KEY END BLOCK VI READ-EDIT". This message is written after the BLOCK VI card-image input has been successfully read and processed. The final message "KEY END INPUT MODULE" indicates that all INPUT module operations are completed.

Page 2 of the output lists the EDIT module input as provided in the BLOCK VI card-image input. Chapter VII provides a detailed description of the BLOCK VI input parameters and the edit quantities produced. Both "point" and "zone" edits are requested. Referring to the card-image input, the points at which edits are desired are provided in the POINTS array input where the mesh points 1 through 10 and 46 through 50 are specified (note the use of the linear interpolate operator described in Ch. III in specifying the POINTS array input). Since no Edit Zones are explicitly specified (no EDZONE array input is specified), the code will assume that the Edit Zones are the same as the Coarse-Mesh intervals specified in Sample Problem 1. Since the parameter IGRPED is input with a

value of zero, only the energy-group totals for each edit quantity are to be printed. Resident macroscopic, resident constituent, and material cross-section response functions are specified using both the $v\Sigma_f$ and Σ_f cross-section types.

Page 3 provides the desired edit output, or reaction rates, for the material "FUEL", $(U,Pu)O_2$, specified in Sample Problem 1 and also for the resident macroscopic cross sections at each spatial mesh point requested. Also provided are the Edit Zone (defaulted to Coarse-Mesh interval) sums requested.

Page 4 provides the desired edit output, or reaction rate information, for the "CONSTITUENTS" PU-239 and U-238 both at the desired space-points and as sums (integrals) over the Edit Zones (coarse-mesh intervals).

Page 5 shows the RUN HIGHLIGHTS and STORAGE/TIMING HISTORY for this sample problem.

GENERALIZED INPUT MODULE RUN ON 11/05/81 WITH VERSION 11-02-81

...LISTING OF CARDS IN THE INPUT STREAM...

```

1      2      0      0
2. SAMPLE PROBLEM 2 FOR USER'S MANUAL
3. EDITS (ONLY) ON PREVIOUS SAMPLE PROBLEM 1
4. / GEOMETRY - FROM PREVIOUS GEDST STANDARD INTERFACE FILE
5. / CROSS SECTIONS - FROM PREVIOUS MACRXS AND SNKEDT INTERFACE FILES
6. / MIXING - FROM PREVIOUS NDXSRF AND ZNATON INTERFACE FILES
7. / SOLVER - NO INFORMATION SUPPLIED. USING PREVIOUS RTFLUX
8. / - FILE FROM SOLVER RUN.
9. / EDITS - CARD INPUT SUPPLIED
10. /
11. /
12. / * * * BLOCK I * * *
13. IGEDM=2, NGROUP=2, ISN=4 NISO=7 MT=4 NZONE=3 IM=3 IT=50 T
14. /
15. /
16. / * * * BLOCK VI (EDITS) * * *
17. PTED=1 ZNED=1 POINTS=811,10 3146.50 IGRPED=0
18. EDAS= NUSIDF FISS RESDNT=1 EDMATS=FUEL
19. EDCGNS= *PU-239* *U-238* 1

```

CASE TITLE

KEY START CASE INPUT

```

2 NHEAD NUMBER OF TITLE CARDS TO FOLLOW
0 NDTTY 0/1 NO/YES SUPPRESS ON-LINE TERMINAL OUTPUT
0 NDLIST 0/1 NO/YES SUPPRESS INPUT LISTING

```

```

* SAMPLE PROBLEM 2 FOR USER'S MANUAL
* EDITS (ONLY) ON PREVIOUS SAMPLE PROBLEM 1

```

KEY END BLOCK I READ

... BLOCK I - CONTROLS AND DIMENSIONS ...

... DIMENSIONS (ARRAY NAME = DIMENS) ...

```

2 IGEDM 1/2/3 PLANE/CYLINDER/SPHERE
2 NGROUP NUMBER OF ENERGY GROUPS
4 ISN ANGLUAR QUADRATURE ORDER
7 NISO NUMBER OF INPUT ISOTOPES (FROM ISDTXS, GRUPXS, OR CARDS)
4 MT NUMBER OF PERMANENT MATERIALS
3 NZONE NUMBER OF ZONES
3 IM NUMBER OF COARSE MESH X INTERVALS
50 IT NUMBER OF FINE MESH X INTERVALS

```

... STORAGE ...

MAXLCM= 50000

KEY END BLOCK VI READ-EDIT

KEY END INPUT MODULE

```

.....
... EDIT OUTPUT ...
.....
BLOCK VI - EDIT SPECIFICATION DATA
.....
*KEY START EDIT OUTPUT *
.....
... INPUT CONTROL INTEGERS...
1 PTED 0/1 NO/YES - POINT EDITS DESIRED
1 ZNEO 0/1 NO/YES - ZONE EDITS DESIRED
0 A*ED 0/1 DIRECT/ADJOINT EDIT(USE RTFLUX/ATFLUX FILE)
0 IGRPED 0/1/2/3 PRINT TOTALS ONLY/PRINT BRDAD GROUPS ONLY/SAME AS 1/PRINT ALL GROUPS AND TOTALS
0 BYVOLP 0/1 NO/YES - MULTIPLY POINT REACTION RATES BY MESH VOLUMES

...ENERGY RELATED EDIT INFORMATION
2 NUMBER OF FINE NEUTRON GROUPS
0 NUMBER OF FINE GAMMA GROUPS
2 TOTAL NUMBER OF FINE GROUPS
2 TOTAL NUMBER OF BRDAD GROUPS

...SPACE RELATED EDIT INFORMATION.
15 NUMBER OF POINTS TO EDIT
3 NUMBER OF ZONES TO EDIT
0 0/1 NO/YES DENSITY FACTORS WERE INPUT
.....

```

```

.....
EDIT OUTPUT
.....
...EDIT SPECIFICATION DATA(CONTINUED)...
.....
...DESCRIPTION OF CROSS SECTION EDITS...
ISOTOPE NO. MATERIAL NO. CONSTITUENT NO.
-----
NDNE- FUEL 2 PU-239 6
RESOINT U-238 7

REACTION RATES WILL BE FORMED FOR EACH
OF THE ABOVE
USING THE CROSS SECTION TYPES SHOWN BELOW

TYPE POSITION
---- NO.
NUSIGF 2
FISS 5
.....

```

.....
 * KEY START MATERIALS *

.....
 * REACTION RATES *
 * FROM *
 * MATERIALS *

.....
 * POINT EDIT FOR THE SUM OF THE GROUPS *

.....
 * PERMANENT MATERIAL 2 - FUEL *

POINT	AV RAD	NUSIGF	FISS
1	1.0000	8.12311E-04	2.66161E-04
2	3.0000	8.09497E-04	2.65237E-04
3	5.0000	8.02380E-04	2.63234E-04
4	7.0000	7.94363E-04	2.60282E-04
5	9.0000	7.82500E-04	2.56398E-04
C	11.0000	7.67878E-04	2.51611E-04
7	13.0000	7.50577E-04	2.45947E-04
8	15.0000	7.30695E-04	2.39438E-04
9	17.0000	7.08341E-04	2.32120E-04
10	19.0000	6.83642E-04	2.24035E-04
46	91.0000	1.69157E-05	5.62451E-06
47	93.0000	1.40864E-05	4.68360E-06
48	95.0000	1.12922E-05	3.75422E-06
49	97.0000	8.48376E-06	2.81997E-06
50	99.0000	5.55269E-06	1.84478E-06

.....
 * RESIDENT MACROSCOPIC *

POINT	AV RAD	NUSIGF	FISS
1	1.0000	2.84309E-04	9.31563E-05
2	3.0000	2.83324E-04	9.28330E-05
3	5.0000	2.81183E-04	9.21319E-05
4	7.0000	2.78027E-04	9.10947E-05
5	9.0000	2.73875E-04	8.97393E-05
6	11.0000	2.68757E-04	8.80638E-05
7	13.0000	2.62702E-04	8.60814E-05
8	15.0000	2.55743E-04	8.38033E-05
9	17.0000	2.47919E-04	8.12421E-05
10	19.0000	2.39275E-04	7.84124E-05
46	91.0000	0.	0.
47	93.0000	0.	0.
48	95.0000	0.	0.
49	97.0000	0.	0.
50	99.0000	0.	0.

.....
 * ZONE EDIT FOR THE SUM OF THE GROUPS *

.....
 * PERMANENT MATERIAL 2 - FUEL *

ZONE	VOLUME	NUSIGF	FISS
1	5.0265E+03	2.76872E+00	9.07859E-01
2	1.0367E+04	1.38690E+00	4.58664E-01
3	1.6022E+04	4.04085E-01	1.34306E-01
SUM	3.1416E+04	4.55971E+00	1.50083E+00

.....
 * RESIDENT MACROSCOPIC *

ZONE	VOLUME	NUSIGF	FISS
1	5.0265E+03	9.69053E-01	3.17751E-01
2	1.0367E+04	2.43492E-02	8.11639E-03
3	1.6022E+04	0.	0.
SUM	3.1416E+04	9.93402E-01	3.25867E-01

.....

.....
 *KEY START CONSTITUENTS
 *.....

```

    *.....*
    *          REACTION RATES          *
    *          FROM                     *
    *          CONSTITUENTS             *
    *.....*
  
```

.. POINT EDIT FOR THE SUM OF THE GROUPS

.. CONSTITUENT ISOTOPE 6 PU 239			
POINT	AV RAD	NUSIGF	FISS
1	1.0000	2.610RJE-04	8.54143E-05
2	3.0000	2.60169E-04	8.51148E-05
3	5.0000	2.58210E-04	8.44743E-05
4	7.0000	2.55324E-04	8.35310E-05
5	9.0000	2.51527E-04	8.22901E-05
6	11.0000	2.46847E-04	8.07605E-05
7	13.0000	2.41310E-04	7.89509E-05
8	15.0000	2.34948E-04	7.68117E-05
9	17.0000	2.27798E-04	7.45349E-05
10	19.0000	2.19900E-04	7.19541E-05
46	91.0000	0	0
47	93.0000	0	0
48	95.0000	0	0
49	97.0000	0	0
50	99.0000	0	0

.. CONSTITUENT ISOTOPE 7 U-238			
POINT	AV RAD	NUSIGF	FISS
1	1.0000	2.32259E-05	7.74197E-06
2	3.0000	2.31546E-05	7.71820E-06
3	5.0000	2.29728E-05	7.65761E-06
4	7.0000	2.27031E-05	7.56770E-06
5	9.0000	2.23478E-05	7.44927E-06
6	11.0000	2.19100E-05	7.30332E-06
7	13.0000	2.13915E-05	7.13051E-06
8	15.0000	2.07947E-05	6.93156E-06
9	17.0000	2.01217E-05	6.70722E-06
10	19.0000	1.93749E-05	6.45829E-06
46	91.0000	0	0
47	93.0000	0	0
48	95.0000	0	0
49	97.0000	0	0
50	99.0000	0	0

.. ZONE EDIT FOR THE SUM OF THE GROUPS

.. CONSTITUENT ISOTOPE 6 PU 239			
ZONE VOLUME	NUSIGF	FISS	
1	5.0265E+03	8.93230E-01	2.92476E-01
2	1.0367E+04	0	0
3	1.6022E+04	0	0

SUM	3.1416E+04	8.93230E-01	2.92476E-01

.. CONSTITUENT ISOTOPE 7 U 238			
ZONE VOLUME	NUSIGF	FISS	
1	5.0265E+03	7.58228E-02	2.52743E-02
2	1.0367E+04	2.43492E-02	8.11639E-03
3	1.6022E+04	0	0

SUM	3.1416E+04	1.00172E-01	3.33907E-02

REFERENCES

1. R. D. O'Dell, "Standard Interface Files and Procedures for Reactor Physics Codes, Version IV," Los Alamos Scientific Laboratory report LA-6941-MS (September 1977).
2. B. M. Carmichael, "Standard Interface Files and Procedures for Reactor Physics Codes, Version III," Los Alamos Scientific Laboratory report LA-5486-MS (February 1974).
3. American National Standard ANSI X3.9-1966, "USA Standard FORTRAN," American National Standards Institute, New York.
4. G. I. Bell and S. Glasstone, "Discrete Ordinates and Discrete S_N Methods," in Nuclear Reactor Theory (Van Nostrand Reinhold, New York, 1970), Chap. 5, pp. 232-235
5. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," Nucl. Sci. Eng. 64, 344 (1977).
6. B. G. Carlson and K. D. Lathrop, "Transport Theory-Method of Discrete Ordinates," in Computing Methods in Reactor Physics, H. Greenspan, C. N. Kelber and D. Okrent, Eds. (Gordon and Breach, New York, 1968), Chap. III, p. 185.
7. Ref. 6, p. 211.