LA-12969-M
MANUAL

UC-705
Issued: June 1995

DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System

Ray E. Alcouffe, Randal S. Baker,
Forrest W. Brinkley, Duane R. Marr,
R. Douglas O'Dell, and Wallace F. Walters*

*(Deceased)

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

An Affirmative Action/Equal Opportunity Employer

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT – Introduction and System Overview

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

1

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

The DANTSYS code package includes the following transport codes: ONEDANT, TWODANT, TWODANT/GQ, TWOHEX, and THREEDANT. This document is the central user, methods and programming documentation for the system of codes.

The DANTSYS code package is a modular computer program package designed to solve the time-independent, multigroup discrete ordinates form of the Boltzmann transport equation in several different geometries. The modular construction of the package separates the input processing, the transport equation solving, and the post processing (or edit) functions into distinct code modules: the Input Module, one or more Solver Modules, and the Edit Module, respectively. The Input and Edit Modules are very general in nature and are common to all the Solver Modules. The ONEDANT Solver Module contains a one-dimensional (slab, cylinder, and sphere), time-independent transport equation solver using the standard diamond-differencing method for space/angle discretization. It was previously documented in Ref. 1. Also included in the package are Solver Modules named TWODANT, TWODANT/GQ, THREEDANT, and TWOHEX. The TWODANT Solver Module solves the time-independent two-dimensional transport equation using the diamond-differencing method for space/angle discretization and was previously documented in Ref. 2. We have also introduced an adaptive weighted diamond differencing (AWDD) method for the spatial and angular discretization into TWODANT as an option. The TWOHEX Solver Module solves the time-independent two-dimensional transport equation on an equilateral triangle spatial mesh. The user's guide for TWOHEX was previously documented in Ref. 3. The THREEDANT Solver Module solves the time independent, three-dimensional transport equation for XYZ and RZΘ symmetries using both diamond differencing with set-to-zero fixup and the AWDD method. The TWODANT/GQ Solver Module solves the two-dimensional transport equation in XY and RZ symmetries using a spatial mesh of arbitrary quadrilaterals. The spatial differencing method is based upon the diamond differencing method with set-to-zero fixup with changes to accommodate the generalized spatial meshing.

This manual describes the standardized Input and Edit Modules together with each of the Solvers in the package. Throughout this manual we will refer to this package as the DANTSYS code package.

Some of the major features included in the DANTSYS code package are:

- a free-field format ASCII text input capability;
- standardized, data- and file-management techniques as defined and developed by the Committee on Computer Coordination (CCCC) and described in Ref. 4; both sequential file and random-access file handling techniques are used;
- the use of a diffusion synthetic acceleration scheme to accelerate the iterative process in the Solver Modules ONEDANT, TWODANT, TWODANT/GQ, and THREEDANT;
- direct (forward) or adjoint calculational capability;

- standard plane, two-angle plane, cylindrical or spherical geometry options for 1-d;
- x-y, r-z, r-theta, equilateral triangular mesh x-y, and generalized quadrilateral x-y and r-z geometries in 2-d;
- x-y-z and r-z-theta geometries in 3-d;
- arbitrary anisotropic scattering order;
- vacuum, reflective, periodic, white, albedo*, or surface source boundary condition options;
- inhomogeneous (fixed) source or k_{eff} calculation options as well as time-absorption (alpha), nuclide concentration, or dimensional search options;
- "diamond-differencing" for solution of the transport equation;
- AWDD available in the TWODANT and THREEDANT modules;
- A coupled S_n/Monte Carlo option for the TWODANT module;
- user flexibility in using both ASCII text or sequential file input;
- user flexibility in controlling the execution of both modules and submodules;
- extensive, user-oriented error diagnostics.

DANTSYS 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 reasonable default values for many input parameters are provided. The input is designed to be meaningful, easily understood, easily verified, easy to change, and logically common for all solvers. The printed output is well documented with liberal use of descriptive comments and headings.

*Only operative in the ONEDANT solver.

DOCUMENTATION SUMMARY

The S_n Code package includes documentation for varying audiences.

Table 1.1 Documentation Elements

Documentation Type	Title
Introduction	Introduction and System Overview
Users Guide	ONEDANT Users Guide
	TWODANT Users Guide
	TWODANT/GQ Users Guide
	TWOHEX Users Guide
	THREEDANT Users Guide
Details	Details of the Geometry and Solver Input
Details	Running the Edit Module
Reference	Free Field Input Reference
Reference	Cross Section Libraries
Details	Material Mixing Tutorial
Methods	Methods Document
Details	Code Structure
Reference	Error Messages
Reference	File Descriptions
Reference	Solver Module Abstracts
Reference	Bibliography

Since the ONEDANT Code Package was first released in 1982, it has undergone numerous changes in the form of bug fixes, modification to improve the code's robustness, and new features and capabilities. In addition, the current code package contains the TWODANT, TWODANT/GQ, THREEDANT, and TWOHEX Solver Modules, as well as the ONEDANT Solver Module.

This documentation contains all of the revised ONEDANT user's manual.[1] Also included are the published manuals for TWODANT[2] and TWOHEX.[3]

Newly written are user's guides for the TWODANT/GQ and THREEDANT codes.

The user's guide for each solver is a separate chapter of this document and is intended to be complete enough that it can be extracted from the document and used as a separate input manual for that solver, with the remainder of the document serving as reference material. Thus, you will find a short summary of the free field input in each of the user's guides which will be sufficient to remind the experienced user of the operators available, but more detail will be found in the chapter focusing on the free field input that is found in the complete document. You will also find a references section that focuses on that particular solver in each of the user's guides.

Although the original ONEDANT manual's detailed focus was on the one-dimensional, discrete ordinates Solver, there was much general information on the structure of the overall package, the manner in which input is supplied by the user and multigroup cross section libraries accepted, the manner in which nuclides are mixed, how edits are performed, etc. These general features that apply to the other Solver Modules as well were accordingly moved to their own chapters in this document.

COMMON MODELING CONCEPTS

Certain modeling concepts are common to more than one solver module. For instance, in the geometry area, the concept of a calculational fine mesh is common to all the modules in the package. The concept of a coarse mesh is common to all but TWOHEX and TWODANT/GQ. The transport solution strategy is generally the same for all modules. These common concepts are discussed in the sections below.

Geometry Concepts

In the specification of geometry and space-variable related input, the user must be familiar with the nomenclature used by DANTSYS. The terms fine mesh, coarse mesh, and zones are defined below for the three orthogonal geometry solvers: ONEDANT, TWODANT and THREEDANT.

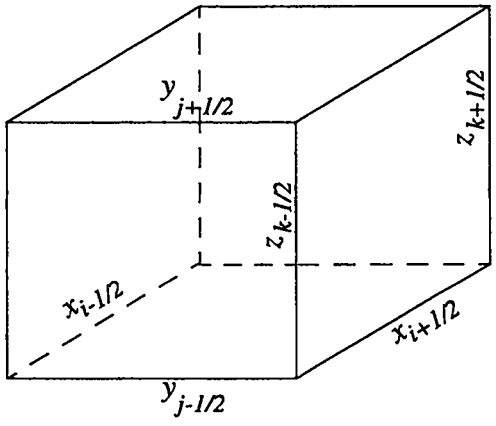


Figure 1.1 Spatial mesh labeling convention in DANTSYS.

The fine mesh is the spatial solution-mesh for the problem, as depicted in Figure 1.1 above. Each fine mesh, or fine mesh interval, is bounded by an adjacent pair of fine-mesh grid-surfaces $x_{i-1/2}$ and $x_{i+1/2}$ with $x_{i-1/2} < x_{i+1/2}$ in the x direction; $y_{j-1/2}$ and $y_{j+1/2}$ with $y_{j-1/2} < y_{j+1/2}$ in the y direction; $z_{k-1/2}$ and $z_{k+1/2}$ with $z_{k-1/2} < z_{k+1/2}$ in the z direction There are IT, JT and KT such fine mesh intervals respectively. No material discontinuities may occur within a fine mesh interval. The specification of the fine mesh is accomplished by specifying how many equally sized fine mesh intervals there are in each 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, JM, and KM coarse mesh intervals in each of the coordinate directions spanning the

problem. Each coarse mesh interval contains one or more fine mesh intervals. All fine mesh intervals within a coarse mesh interval have equal widths. No material discontinuities may occur within a 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. The user assigns a zone number to each coarse mesh interval. The zone number tells the code which macroscopic cross section set is to be used within that zone. Coarse mesh intervals having the same zone number need not be simply connected.

A zone number of 0 (zero) can be used to specify that a coarse mesh interval is a pure void (all cross sections are identically zero).

More detail on these concepts is to be found on page 7-13.

Iteration Strategy

In this section is described the basic iteration strategy used in the execution of all the Solver Modules. A more detailed description of the strategy is given on page 7-15 including the iteration controls that the user inputs and a discussion of the iteration monitor printout that is printed in the output.

The basic features of the iteration strategy are shown in the simplified flow diagram in Figure 1.2.

The iterative strategy is basically divided into three parts: inner iterations, outer iterations, and eigenvalue search iterations.

The inner iterations are concerned with the convergence of the pointwise scalar fluxes in each group due to iteration on within-group scattering processes. For eigenvalue problems, the source to each group is given by the fission source from the previous outer iteration plus any in-scattering sources. For fixed source problems, the source to each group is the input source distribution plus the in-scattering source.

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.

The eigenvalue search iteration is the ability of the code to adjust some parameters of the problem, namely the isotopic concentrations or the spatial dimensions of selected coarse mesh intervals, to obtain a desired value of the k_{eff}. Also the alpha eigenvalue (time constant) of the system is determined by a search procedure based upon successive determinations of k_{eff}.

Both the inner and outer iterations are accelerated using the diffusion synthetic method. See page 12-14 for the theory of this method. TWOHEX does not use the diffusion synthetic acceleration method; rather, it uses Chebychev acceleration.

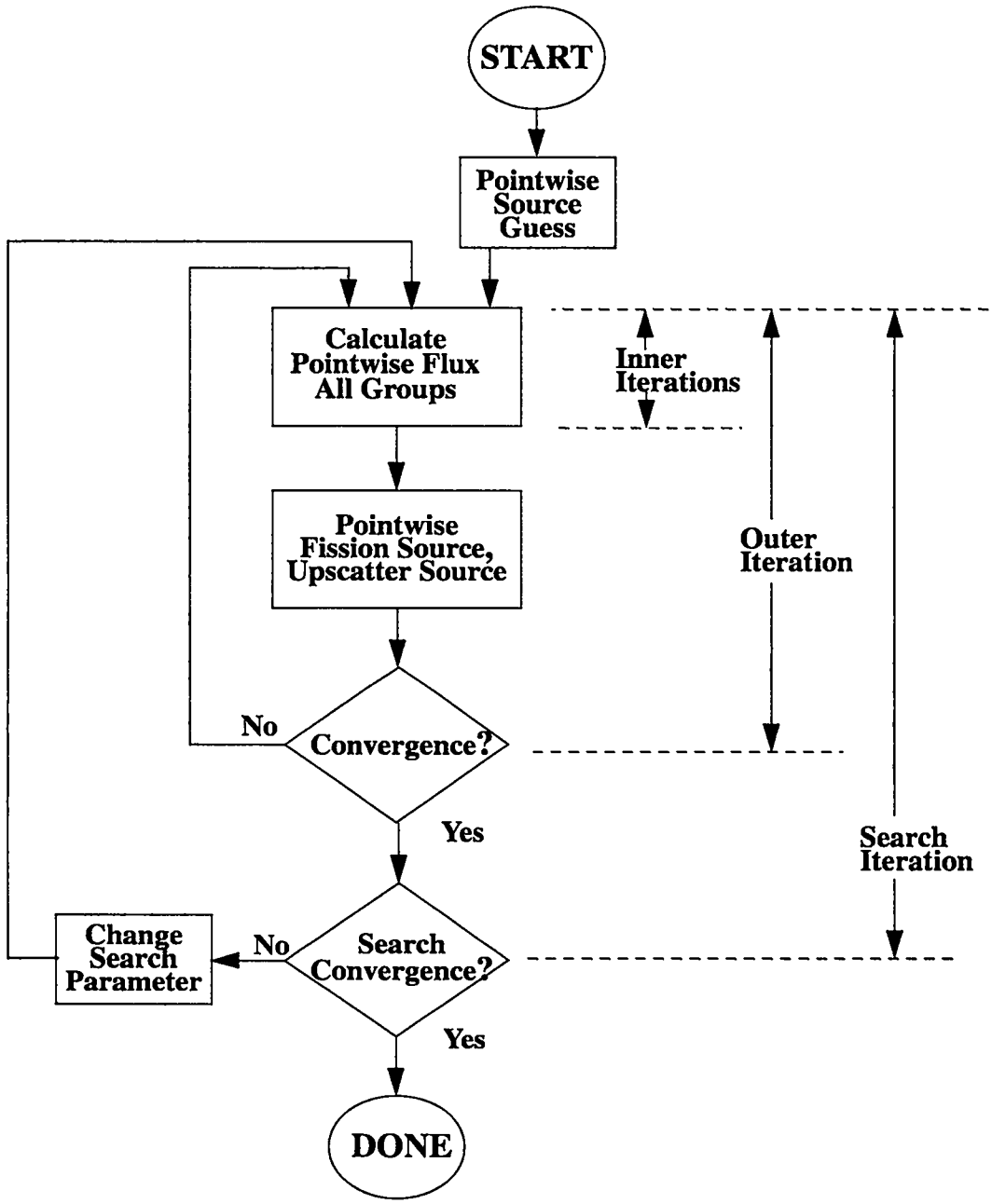


Figure 1.2 Simplified flow diagram of SOLVER iteration strategy

Modular Structure and Interface Files

There are some times when the user must be aware that the code is constructed on a modular basis with files supplying the communication between modules. This general structure is shown in Figure 1.3. Each of the solution techniques employs a different solver module.

The purpose of the input module is to convert the text input into interface files that then drive the solver and optionally the edit modules.

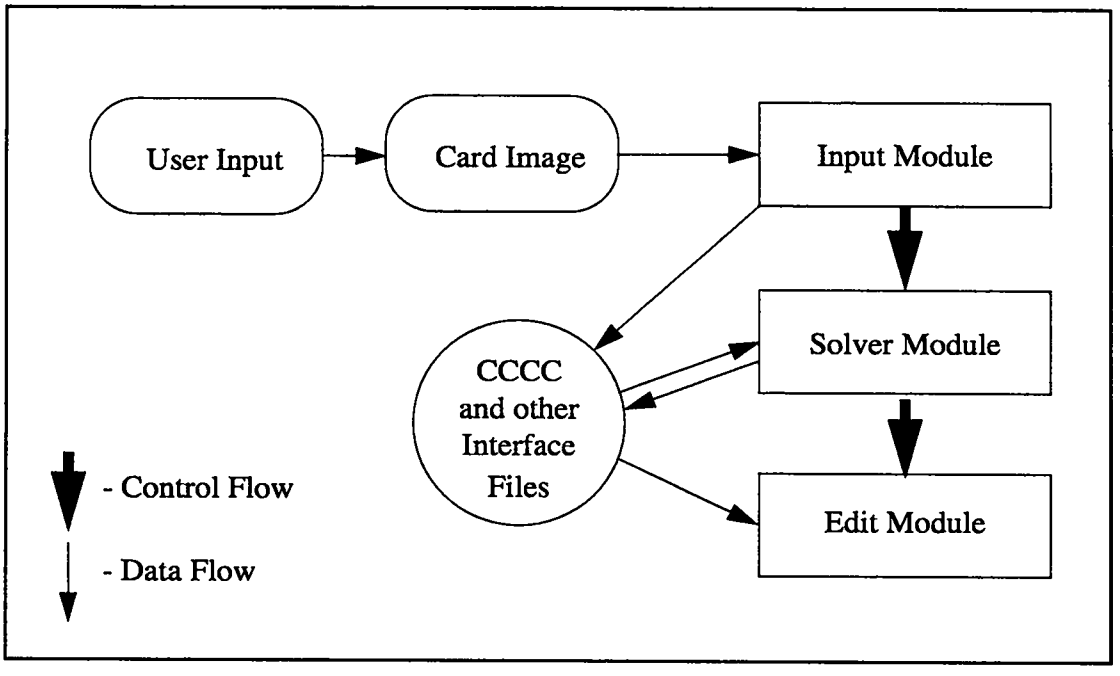


Figure 1.3 DANTSYS Structure

More detail on this topic will be found in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure".

REFERENCES

1. R. D. O'Dell, F. W. Brinkley Jr., D. R. Marr, R. E. Alcouffe, "Revised User's Manual for ONEDANT: A Code Package for One-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory manual LA-9184-M, Rev. (December 1989).
2. R. E. Alcouffe, F. W. Brinkley, D. R. Marr, and R. D. O'Dell, "User's Guide for TWODANT: A Code Package for Two-Dimension, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory manual LA-10049-M, Rev. 1, (October 1984).
3. W. F. Walters, F. W. Brinkley, and D. R. Marr, "User's Guide for TWOHEX: A Code Package for Two-Dimensional, Neutral-Particle Transport in Equilateral Triangular Meshes," Los Alamos National Laboratory manual LA-10258-M, (October 1984).
4. 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).

ONEDANT USER'S GUIDE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

2

XTM — Transport Methods Group

Los Alamos
National Laboratory

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

An Affirmative Action/Equal Opportunity Employer

This work was supported by the US Department of Energy.

USER'S GUIDE FOR ONEDANT: A CODE PACKAGE FOR ONE-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT

by
Ray E. Alcouffe, Forrest W. Brinkley,
Duane R. Marr, and R. Douglas O'Dell

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The ONEDANT code is a modular computer program designed to solve the one-dimensional, time-independent, multigroup discrete-ordinates form of the Boltzmann transport equation.[1,2]

ONEDANT™ is based on the modular construction of the DANTSYS™ code system package. This modular construction separates the input processing, the transport equation solving, and the postprocessing, 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 used by all the codes in the package. Different solution techniques are invoked simply by executing different SOLVER modules in the package. This SOLVER choice is automatically made by the INPUT module through an analysis of the input stream.

The ONEDANT code is simply the DANTSYS code package with a one-dimensional SOLVER module.

Some of the major features included in the ONEDANT package are:

1. a free-field format text input capability, designed with the user in mind,
2. standardized data and file management techniques as defined[3] and developed by the Committee on Computer Code Coordination (CCCC); both sequential file and random-access file handling techniques are used,
3. the use of a diffusion synthetic acceleration scheme[4] to accelerate the iterative process in the SOLVER module,
4. direct (forward) or adjoint calculational capability,
5. standard plane, two-angle, 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 (α), nuclide concentration, or dimensional search options,

9. "diamond-differencing" for solution of the transport equation,
10. user flexibility in using either ASCII text or sequential file input,
11. user flexibility in controlling the execution of both modules and submodules, and
12. extensive, user-oriented error diagnostics.

DOCUMENTATION FOR ONEDANT USAGE

The documentation described here constitutes a complete manual for the use of the ONEDANT code. It is intended to fully replace the former ONEDANT manual.[5]

Included are two general categories of information. The first category is in this User's Guide and is oriented towards preparing input to the code. The second category is of a background, reference, conceptual, tutorial, or theoretical nature and is intended primarily for the novice or first time user; an experienced user generally needs only this User's Guide.

What Is In This User's Guide

This User's Guide is a chapter from the much larger DANTSYS document. This Guide provides the ASCII text input specifications for ONEDANT.

The guide is intended to serve as a complete input manual for two classes of user. Special, succinct sections containing summaries and compact tables are intended for the advanced user in order to make his input preparation more efficient. The main body of the guide concerns itself with descriptions of the input and should be sufficient for the user familiar with discrete ordinates concepts. Novice users may find other chapters of the document necessary.

This Guide first gives an overview of the input block order required by the code.

Next is a "mini-manual" in which are listed all the names of available input arrays arranged by input block. Definitions of input arrays are not given, as the names are suggestive, but expected types and sizes are provided. This mini-manual is very useful to the user as a quick check for completeness, a quick reference to type and size, and an index into the more detailed array descriptions that follow. For the experienced user, the mini-manual is frequently all that is needed to prepare a complete input deck.

Following the mini-manual are reference sections describing in detail all the input parameters and arrays.

Appendix A provides two sample ONEDANT problems with explanation of the output for the user.

Lastly, Appendix B details operating system specifics, including how to effect an execution of the code.

Information of a reference, background, or theoretical nature that the first time user may need may not be found in this User's Guide, but the user will encounter liberal references to other chapters of this document for that sort of information.

What Is Available Elsewhere

In addition to this User's Guide, the user, especially the first time user, may find the information below described in other chapters of this document pertinent. For even greater detail on some of the general items, particularly the methods items, the user should look at Ref. 6.

The chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1 discusses in more detail the geometry and solver concepts and their related input. If the User's Guide proves insufficient for your needs, look in this chapter. Among the many sections of the chapter are ones on the input of inhomogeneous sources and a discussion of eigenvalue searches. There is also more detail on the Block-I input.

A discussion of how the EDIT module works and more detail on preparing the input is given in the chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 serves as the reference manual for the free-field input (rules, format, and operators) used in this code. That chapter is summarized in this guide, but should the summary prove inadequate, the user is referred there for full details.

The chapter "CROSS-SECTION LIBRARIES" starting on page 10-1 gives details of the many library formats available to ONEDANT, including sections on how to prepare your own card-image (or text) libraries.

The chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1 describes the mixing concepts in detail and shows some examples.

Next is the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. That chapter describes the theoretical basis for the ONEDANT code as well as the other codes in the DANTSYS package.

In the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure" starting on page 13-1 is shown a brief 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. In particular, the code package structure must be understood in order to make up input for piecewise executions of the code that are possible with controls that are part of the input in Block-I.

Error diagnostics that the user might encounter are found in the chapter "ERROR MESSAGES" starting on page 14-1. Several examples of input errors and the resulting error messages are provided for the user.

The chapter "FILE DESCRIPTIONS" starting on page 15-1 is a reference that describes all the files used by the package. Included is a detailed description of the file structure of the code dependent, binary, sequential interface files generated by and used in the

DANTSYS package. Also included are descriptions of any other files produced or used by the package, both binary and text. In some cases, this may simply be a reference to a more comprehensive document, such as the file descriptions for the CCCC standard interface files.

ONEDANT INPUT OVERVIEW

Input Block Order

The full ONEDANT input consists of a title line section, followed by six blocks of free field input. The title line section is not free field. Any input referred to as a block uses the free field input form.

Block-I consists of basic control and dimensional information that allows efficient packing of the array data in the available memory. This information also allows checking of the lengths of arrays supplied in subsequent blocks or those from interface files.

Block-II contains the geometric information.

Block-III consists of the nuclear data specifications.

Block-IV contains mixing information.

Block-V contains the rest of the input needed for specifying the flux calculation.

And lastly, Block-VI contains the edit (i.e., report writing) specifications.

If a text cross-section library is to be included in the input deck, it should be placed between Blocks III and IV. ONEDANT supports many library formats and so the library may or may not be in free field format depending upon the option chosen.

A full input would then look like that diagrammed on the following page.

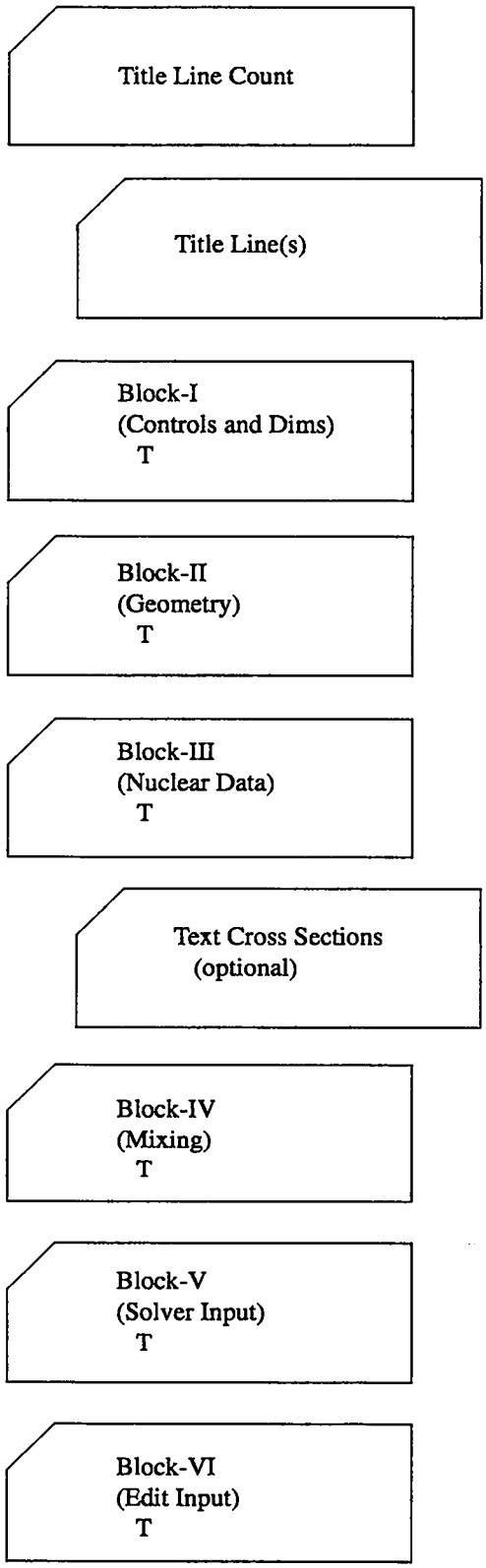


Figure 2.1 ONEDANT Input Order

Free Field Input Summary

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 is summarized here for quick reference.

There are four basic input quantities in the free field input used in ONEDANT; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is briefly described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array." To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, might look like:

SOURCX= 0 0 0 1.1 1.1 0 0 0 0 0

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Single value input variables are treated as arrays of unit length.

Numeric Data Items

Numeric data items follow a Fortran input convention. For example, all of the following are valid entries for the number ten:

10, 1.0+1, 1E1, 10.0

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a Fortran variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters and blanks may be included if the data item is surrounded by double quotes.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this, the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described later, in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read, no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input.

The data operators are specified in the general form

n *O* d

where:

n is the "data numerator," either an integer or a blank;
O is any one of the "data operator" characters shown below; and
d is a "data entry" (may be blank for some operators).

A "data entry" must be a numeric data item; a character data item cannot be used with an operator.

Note: The "data operator" character must be appended to the "data numerator."

Using operators, the SOURCX input described above could more succinctly be given as:

SOURCX= 0 0 0 2R 1.1 F0

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO[7] operators may be used in numeric entry.

A table of the most used operators is given next including brief descriptions. For full descriptions of these and a complete list of all the available operators, including the more esoteric ones, the user is referred to "FREE FIELD INPUT DETAILS" on page 9-13.

Frequently Used Operators

Operator[a]	Functionality
nR d	REPEAT the data item d, n times.
nI d	INTERPOLATE (linear) n data items between data item d and the next data item.
nC d	SCALE (multiply) the n previous entries by d.
F d	FILL the rest of the data string with the data item d.
nY m	STRING REPEAT. Repeat the previous m strings, n times.
nL d	INTERPOLATE LOGARITHMICALLY n data items between d and the next d.
nZ	ZERO. Enter the value zero n successive times.
nS	SKIP. Skip the next n data items.
nQ m	SEQUENCE REPEAT. Enter the last m entries, n more times.
nG m	SEQUENCE REPEAT WITH SIGN CHANGE. Same as the Q option but the sign of the m entries is changed every repeat.
nN m	SEQUENCE REPEAT INVERT. Same as the Q option but the order of the m entries is inverted each repeat.
nM m	SEQUENCE REPEAT INVERT WITH SIGN CHANGE. Same as N option but the sign is also changed every repeat.
nX	COUNT CHECK. Causes code to check the number of entries in the current string so far, against the number n.

a. The operator character must always be appended directly to n. d or m need not be immediately adjacent to the operator character.

MINI-MANUAL Introduction

On the following few pages is given a complete list of the input names, expected array sizes, and order within the array. No description of the array contents is given in this MINI-MANUAL as full details are given in later sections. The MINI-MANUAL is intended to serve as a quick reference for the knowledgeable user.

In both the MINI-MANUAL and in the detailed sections which follow, a shorthand form is used to indicate the size and order of the array that the code expects. This information is enclosed in square brackets immediately after the array name. Essential features are:

1. A single entry in the brackets is the array length.

2. No brackets at all indicates a simple variable (i.e., an array of unit length).

3. A dash (-) in the brackets indicates an arbitrary length.

4. A semicolon (;) indicates that the input for that array is expected in strings. To the left of the semicolon is the string length. To the right of the semicolon is the number of strings in the array.

5. If the number of strings is shown as a product, the order is important. The leftmost quantity must be exhausted first, then the next one to the right is varied. For example, the array name for the full spatial source distribution is shown as:

   SOURCF [IT;NGROUP*NMQ]

   where IT is the number of fine meshes in the X-direction, NGROUP is the number of energy groups, and NMQ is the number of input source moments. For this array, the first string is composed of the P_0 source values for each x mesh point for group 1. The next string is the P_0 source values for each x mesh point for group 2. When NGROUP strings have been entered (thus exhausting the P_0 values), one enters in the next string the P_1 source values for each x mesh for group 1. The P_1 values for group 2 follow. Continue until all NMQ moments are specified.

Note: Usually, values for the quantities within brackets will have already been specified in the input. Sometimes, however, a quantity is derived from the array input itself. For instance, in this particular case, NMQ is not an input quantity; rather, the code counts the number of strings and then, knowing NGROUP, deduces what NMQ must have been.

MINI MANUAL

Title Line Control
(3I6 Format)
NHEAD, NOTTY, NOLIST

Title Line(s)

(IF NHEAD>0)

Block-I: Controls & Dimensions

IGEOM
NGROUP
ISN
NISO
MT
NZONE
IM
IT

MAXLCM
MAXSCM
-- -- -- -- --
MINIPRT

NOFGEN
NOSOLV
NOEDIT

NOGEOD
NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM
T

Block-II: Geometry

XMESH [IM+1]

XINTS [IM]

ZONES [IM]
T

Block-III: Cross Sections

LIB
valid: *ODNINP*
XSLIB
ISOTXS
GRUPXS
BXSLIB
MACRXS
MACBCD
XSLIBB
(local)*MENDF*
(local)*MENDFG*
alternate XSLIB name

WRITMXS
valid: *MACBCD*
XSLIBB
XSLIBF
XSLIBE

LNG
BALXS
NTICHI
CHIVEC [NGROUP]
LIBNAME
-- -- -- -- -- -- --
Rest of this block is needed only for text libraries.

MAXORD
IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD (default:1)
NAMES [NISO]
EDNAME [IHT-3]
NTPI [NISO]
VEL [NGROUP]
EBOUND [NGROUP+1]

T

iff LIB= ODNINP, insert
ASCII text cross sections here

Block-IV: Mixing

MATLS [-;MT]
ASSIGN [-;NZONE]

PREMIX [-;-]

ASGMOD [-;-]
CMOD

MATNAM [MT]
ZONNAM [NZONE]
MATSPEC [-]
valid: *ATFRAC*
WTFRAC
ATDEN

ATWT [-]

T

Block-V: Solver

IEVT
ISCT
ITH
IBL
IBR

EPSI
EPSO
IITL
IITM
OITM
ITLIM

KCALC

Solver (continued)

--- Output Controls ---

FLUXP
XSECTP
FISSRP
SOURCP
GEOMP
ANGP
BALP
RAFLUX
RMFLUX

--- Miscellaneous ---

BWTH
BHGT
NORM
I2ANG

TRCOR
valid: *DIAG*
BHS
CESARO
NO

CHI [NGROUP;M]

DEN [IT]

--- Quadrature -------

IQUAD
WGT [MM]
MU [MM]

--- Flux Guess -------

INFLUX

Solver (continued)

--- Searches ---------

IPVT
PV
EV
EVM
XLAL
XLAH
XLAX
POD

RM [IM]

---- Albedoes ----

LBEDO [NGROUP]
RBEDO [NGROUP]

----Volumetric Source----

INSORS

SOURCE [NGROUP;NMQ]
-or-
SOURCX [IT;NMQ]
-or both-
SOURCX [IT;NMQ] and
SOURCE [NGROUP;NMQ]
-or-
SOURCF [IT;NGROUP*NMQ]

-----Boundary Source-----

SILEFT [NGROUP]
SIRITE [NGROUP]
-or-

SALEFT [MM/2;NGROUP]
SARITE [MM/2;NGROUP]

T

Block-VI: EDIT

PTED
ZNED

POINTS [K], K≤IT
EDZONE [IT]

EDXS [K], K≤NEDT
RESDNT
EDISOS [K], K≤NISO
EDCONS [K], K≤NISO
EDMATS [K], K≤MT
XDF [IT]

RSFE [NGROUP;-]
RSFX [IT;-]
RSFNAM [-]

ICOLL [K], K≤NGROUP
IGRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

T

ONEDANT INPUT DETAILS

Introduction

The following pages of this section give details for each of the input arrays. All valid ONEDANT arrays are discussed in this section in detail complete enough to form the input.

However, the beginning user, particularly one unfamiliar with discrete-ordinates codes, may find that he is missing some information of a background nature. See "What Is Available Elsewhere" on page 2-14 for that.

First, here are a few general instructions:

1. All six of the input blocks are normally included. Block-I is always required but any of the other five blocks may be omitted under the proper conditions. The input module reads each block in turn and from it generates one or more binary interface files. The interface files drive the SOLVER and EDIT modules. Thus, if the user wants no edits, the Block-VI input may be omitted. Then with no interface file, the EDIT module will not be executed. Alternatively, if the interface file is available from another source, the corresponding block of input may be omitted. For instance, Block-II describes the geometry. The input module normally writes this information to the GEODST interface file. If the GEODST file is available from another source or a previous run, the Block-II input may be omitted.

2. A general theme of the ONEDANT input is that arrays that are not needed are not entered. Presence of an array indicates that it should be used. Thus, for example, if the density array is entered (DEN array), the cross section at each mesh interval will be modified accordingly. No separate switch need be set to say that the calculation should be done. To eliminate the density modification, simply remove the DEN array from the input or comment it out.

3. The arrays, in general, are grouped in the input instructions according to function. Thus, for example, the input arrays for the volumetric source are found in a single table, or grouping, of input.

4. Groupings of input data may be marked as "Required" or "Optional" in order to guide the user and speed navigation through the input instructions.

   "Required" means that at least one of the arrays in the grouping must be entered. Thus, you must read through the grouping and enter at least one of the arrays found there.

Groupings marked "Optional" may be skipped if the subject is inappropriate. Thus, using the previous example, if one has no volumetric source, one simply skips to the next grouping of input; there is no need to read about any of the arrays within the volumetric source grouping.

Arrays in groupings not marked as "Required" or "Optional" should be reviewed. These groupings contain arrays of vital data that are used in every calculation, but have default values. Thus, although you may not make any input to these arrays and they are in that sense optional, you must concern yourself with them to ensure that the default values are what is intended.

5. Input arrays may also be marked individually. If not marked, they inherit the marking of the grouping in which they are contained. Thus, an unmarked array in a "Required" grouping is required input and you must enter that array. An unmarked array in an "Optional" grouping is optional.

   You may encounter a "Required" array within an "Optional" grouping. That means that if you decide to invoke the option represented by that grouping, you must input that particular array. For example, if you want user defined response function reaction rates calculated, you must input the RSFE array.

   All arrays within unmarked groupings are optional. However, values in these arrays may be used by the code, so you should concern yourself with the default values if you choose not to enter a value.

6. Unless specifically noted otherwise, the default on all numeric inputs is zero.

7. In an adjoint run, none of the groupwise input arrays should be inverted. The code will externally identify all groups by the physical group number, not by the calculational group number (the calculational group number is in inverse order). Thus, the user interface should be consistently in the physical group order.

8. The use of information within square brackets to indicate the size of arrays and strings and the order within those arrays is the same as described in "MINI-MANUAL Introduction" on page 2-22.

9. Except where noted, arrays and strings must contain the exact number expected by the code (as indicated in the array or string description). If not, the code will eventually abort with a (hopefully) descriptive error message or messages.

10. New users reading these instructions for the first time and unfamiliar with the ONEDANT input may find it helpful to follow the sample input in Appendix A while reading this section.

11. Array names are shown here in upper case. What you should actually input for them will depend upon the code's implementation on your platform. At the present time, on most platforms, you should use lower case input.

12. Items in italics in the input instructions indicate actual values that may be entered for an array. You will frequently find switches where the input is the digit 0 or the digit 1. This will be represented by *0/1* in the input description. In other arrays where an exact character string is required such as "ISOTXS" in the LIB array, you will find the notation *ISOTXS*. Note that in this notation, the word is both upper case and italicized. This combination means you must enter exactly those characters. Again, although the characters will be shown here in upper case, what you should actually input for them will depend upon the code's implementation on your platform.

13. When a template for the input form is given, as for the MATLS array, the style in the template tells the user what is expected. If an input word or value is lower case and italicized, the user is to replace that position with the entry of his choice. If the input word is in italicized style and in upper case, the user is to input exactly those characters to achieve the desired result. Depending on the implementation on your platform, the input word, itself, is usually in lower case.

14. Units to be used for the input quantities are not spelled out as they only need to be self consistent. However, the following are commonly used: Dimensions in centimeters, isotopic cross sections in barns per atom; then it follows that atom densities are in atoms per barn-centimeter. Sources are particles per cm^3 per second for volumetric sources and particles per cm^2 per second for boundary sources; fluxes will then be in particles per cm^2 per second.

Title Line Details

Title Line Control (format 3I6)[a] {Required}

Word	Name	Comments
1	NHEAD	Number of title lines that follow.[b]
2	NOTTY	Suppress output to on-line user terminal? *0/1* = no/yes.
3	NOLIST	Suppress listing of all ASCII text input? *0/1* = no/yes. (default=no)

a. WARNING! Note that this first line is in fixed format.

b. Follow this control line with NHEAD title lines containing descriptive comments. Each title line may contain up to 72 characters.

Block-I Details: Dimensions and Controls

Dimensions {Required}

Name	Comments
IGEOM	Geometry. *1/2/3* = planar/cylindrical/spherical. or use one of the following character strings: *PLANE, CYLINDER, CYL, SPHERE, SPH, SLAB, SLAB2ANG*[a]
NGROUP	Number of energy groups.
ISN	S_n order to be used. Must be even.
NISO	Number of isotopes on the basic input cross-section library.
MT	Number of materials[b] in the problem(mixed from the isotopes).
NZONE	Number of geometric zones[c] in the problem.
IM	Total number of coarse mesh intervals[d] in the X (or R) direction.
IT	Total number of fine mesh intervals[e] in the X (or R) direction.

a. If SLAB2ANG is entered, the problem will be run as a 2-angle plane calculation and the Block-V parameter I2ANG need not be entered.

b. Material is defined on page 2-41.

c. Zone is defined on page 7-13.

d. Coarse mesh is defined on page 7-13.

e. Fine mesh is defined on page 7-13.

Storage Requirements {Optional}

Name	Comments
MAXSCM	Length of SCM desired (default=40000_{10})
MAXLCM	Length of LCM desired (default=140000_{10})

Note: The above input (Dimensions plus Storage Requirements) for Block-I will cause the code to attempt to produce a full run, subject to availability of the input normally found in the other Blocks. The controls below allow shortened print files, partial runs (say, of only the input module), or cause the code to ignore any of the other input Blocks present. For full details on their use, see "PIECEWISE EXECUTION" on page 13-19.

Minimum Print Output {Optional}

Name	Comments
MINIPRT	Provide a shortened output print file.[a] *0/1 = no/yes.*[b]

a. This shortened print file contains the title, the name of the cross-section library used, the core storage needed, the iteration monitor, the global balance table, and not much else. However, existing print controls, such as the variable NOLIST on the first line which controls the listing of the input lines and the array print controls in the solver modules, work as before and are not affected by the MINIPRT input. The EDIT module also works as before; MINIPRT has no effect on its output.

b. You may also use the words no or yes as an entry.

Run Configuration Controls {Optional}

Name	Comments
NOSOLV	Suppress solver module execution. *0/1* = no/yes.
NOEDIT	Suppress edit module execution. *0/1* = no/yes.
NOGEOD	Suppress writing GEODST file even though the geometry input (Block-II) may be present. *0/1* = no/yes.
NOMIX	Suppress writing mixing files even though the mixing input in Block-IV may be present. *0/1* = no/yes.
NOASG	Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. *0/1* = no/yes.
NOMACR	Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. *0/1* = no/yes.
NOSLNP	Suppress writing the SOLINP file even though Block-V may be present. *0/1* = no/yes.
NOEDTT	Suppress writing the EDITIT file even though Block-VI may be present. *0/1* = no/yes.
NOADJM	Suppress writing the ADJMAC file even though an adjoint calculation is called for. *0/1* = no/yes.

Note: Default on all these controls is no.

Block-II Details: Geometry

Geometry Arrays[a] {Required}

Name	Comments
XMESH [IM+1]	x (or r) coordinates of coarse mesh edges.
XINTS [IM]	Number of fine meshes in each coarse x (or r) mesh. (Entries must sum to IT).
ZONES [IM]	Zone number[b] for each coarse mesh. This array defines the geometric zones to which cross-section materials are assigned. The zone number must not be greater than NZONE.

a. Definitions of coarse mesh, fine mesh, and zone are given in the chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1. The information entered in this block is written to the CCCC standard interface file GEODST.

b. A zone number of zero indicates the mesh contains a void, and no cross section will be associated with that mesh. The zero zone number is not counted in the total zone count NZONE.

Block-III Details: Nuclear Data

Nuclear Data Type and Options {Required}

Name	Comments
LIB	Name[a] and form of the cross-section data file. Enter as a data item one of the following words:

Word	Description
ISOTXS[b]	CCCC standard isotope ordered binary cross-section file.
XSLIB	ASCII text library supplied in a separate file named XSLIB.
ODNINP	ASCII text library follows after this block of input (after the T of Block-III).
GRUPXS[c]	CCCC standard group ordered cross-section file.
BXSLIB	Binary library supplied as a separate file named BXSLIB. [See "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 10-12.
MACRXS[d]	Use existing files named MACRXS for SOLVER module, SNXEDT for EDIT module. These files were created in a previous run. 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.
XSLIBB	See "XSLIBB Card-Image Library File" on page 10-12.
MACBCD	ASCII form of MACRXS file.
MENDF	(LANL only) See "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.
MENDFG	(LANL only) See "The Los Alamos MENDF5G Gamma Cross-Section Library" on page 10-14.
other	If a word other than those listed above is entered, the code will use the file with that word as its name, provided that file exists in the user's file space. Such a file must be structured as an XSLIB file.

Nuclear Data Type and Options (Cont.) {Required}

Name	Comments
WRITMXS {optional}	Controls the code's writing certain ASCII cross-section files.[e] Enter one of the following words: Word — Description *MACBCD* — Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file. *XSLIBB* — Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file. *XSLIBE* — Creates the cross-section file named XSLIBE, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBE is in Los Alamos 6E12 format (IFIDO=0). *XSLIBF* — Creates the cross-section file named XSLIBF, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBF is in FIDO fixed-field format (IFIDO=1).
LNG {optional}	Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.
BALXS {optional}	cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using! Value — Description *-1* — balance cross sections by adjusting absorption cross section. *0* — do not balance cross sections. (default) *1* — balance cross sections by adjusting self-scattering cross section.
NTICHI {optional}	MENDF fission fraction to be used for the problem (LANL only). *1/2/3* = Pu239/U235/U238 (default is U235). Will be overridden by any CHIVEC input described below or by any zone-dependent CHI in input Block-V.
CHIVEC [NGROUP] {optional}	Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.

a. On UNIX systems, the user may specify a search path for some of these files using the environment variable SNXSPATH. See "Library Search Path" on page 2-94 for details.

b. The CCCC standard for file ISOTXS does not allow the inclusion of the 2L+1 term in the higher order scattering cross section. However, if you have a nonstandard file which contains the 2L+1 term, you may override by setting I2LP1=1. See "Text Cross-Section Library Format" on page 2-39. ONEDANT will then convert the cross sections to the appropriate internal form.

c. The 2L+1 term on GRUPXS is treated the same as for ISOTXS. See footnote b..

d. In the convention used in this user's guide, a MACRXS library and its image MACBCD contain "material" cross sections; all the other libraries contain "isotope" cross sections.

e. See "COUPLED NEUTRON-GAMMA CROSS SECTIONS" on page 10-15.

Alternate Library Name {Optional}

Name	Comments
LIBNAME	Alternate name of the library file. May be used only with certain types of libraries. See Table 2.1.

The entries in the LIB input variable normally dictate both the form and the name of the cross-section library. If the user specified ISOTXS, for example, the code would look for a file named ISOTXS and expect it to be in the CCCC format for an ISOTXS file.

For some libraries, the user may specify the form in the LIB array and specify separately the name in the LIBNAME array. The libraries that can be treated this way are shown in .

Table 2.1 LIBNAME Availability

LIB	LIBNAME AVAILABLE?
MACRXS	No
GRUPXS	Yes
ISOTXS	Yes
BXSLIB	Yes
ODNINP	No
MACBCD	No
XSLIBB	No
MENDF[a]	No
MENDFG[b]	No
XSLIB	Yes
other	Ignored

a. Available only at Los Alamos.

b. Available only at Los Alamos.

The BXSLIB file requires special treatment. It is normally created when the original library is a text library in the ODNINP or XSLIB form. In subsequent runs, this binary BXSLIB file may be used as the source of the cross-section data. The user may wish to save this file under another name. The program, in future runs, may then access the library for reading by using LIBNAME to specify that name.

This procedure is wise because some cases using the BXSLIB form as input also require rewriting it in order to add new information. When this situation arises, the rewritten file is always named BXSLIB. Thus, if the original BXSLIB form library had a different name, it would be protected from being overwritten. For the remainder of the current run, the program will access the file named BXSLIB

Text Cross-Section Library Format {Required if LIB= XSLIB or LIB=ODNINP}

Name	Comments
MAXORD	Highest Legendre order in the scattering tables.
IHM	Number of positions (entries) in each row of the cross-section table.
IHT	Position number of the total cross section.
IHS {optional}	Position number of the self-scatter cross section. (default = IHT + 1).
IFIDO {optional}	Format of the cross-section library. *-1/0/1/2* = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.
ITITL {optional}	A title line precedes each table. *0/1* = no/yes
I2LP1 {optional}	Higher order scattering cross sections on the library contain the 2L+1 term. *0/1* = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.
SAVBXS {optional}	Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. *0/1* = no/yes.
KWIKRD {optional}	Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? *0/1* = no/yes (default=yes).
NAMES [NISO] {optional}	Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . .etc.).
EDNAME [IHT-3] {optional}	Character name for each of the EDIT cross-section positions used in the cross-section edits. These are the positions before the absorption cross section in the cross-section table. (default names are: EDIT1, EDIT2, . . .etc.).
NTPI [NISO] {optional}	Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).
VEL [NGROUP] {optional}	Speeds for each group. Needed only for alpha calculations.
EBOUND [NGROUP+1] {optional}	Energy group boundaries.

ASCII text libraries may be entered in one of the four forms indicated by the IFIDO input. All four forms share the following features: Cross sections are entered in a table optionally preceded by a title line. A table consists of NGROUP rows of entries. Each row contains the cross sections for a single group and consists of IHM entries. The user specifies the positions in the row occupied by the total and selfscattering cross sections. Order within a row (e.g., for group g) is then as follows:

$$\ldots \sigma_{abs},\ \nu\sigma_{f},\ \sigma_{total},\ \ldots \sigma_{g+2\rightarrow g},\ \sigma_{g+1\rightarrow g},\ \sigma_{g\rightarrow g},\ \sigma_{g-1\rightarrow g},\ \sigma_{g-2\rightarrow g},\ \text{etc.}$$

Notice that all terms in the scattering matrix are in positions relative to that of the self-scattering position and the rest of the cross sections are in positions relative to the position of the total cross section. The positions before the absorption cross section are frequently used for edit cross sections. For more detail, see "Ordering of Cross Sections Within a Cross-Section Table" on page 10-10.

Different Legendre orders are in different tables, which follow in order.

The user may order the group structure either by increasing energy or by decreasing energy. However, it is conventional and desirable for most problems to order it by decreasing energy, that is, group one is the highest energy. In that case, the scattering cross sections to the left of $\sigma_{g\rightarrow g}$ such as $\sigma_{g+1\rightarrow g}$ are upscattering terms and the terms to the right of $\sigma_{g\rightarrow g}$ are the downscattering terms.

In the Los Alamos format, the table is entered with a standard Fortran 6E12 format.

For greater precision in your input, use the 4E18 option.

In the fixed field FIDO format that ANISN[7] uses, entries are made in six twelve-column fields. Each twelve-column field is divided into three subfields, a two-column numeric field, a one-column character field, and a nine-column numeric field. See page 9-19 for details if you are not familiar with this input. The last field in each table must have the character T in the character position. No array identifier should be used. This format also restricts the usable input operators to T, *, R, -, +, and Z.

In the free field form, entries do not have to be in designated columns. Rather, the rules specified in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 apply. Each table in this form is also terminated with the character T. No array identifier (i.e., array name with appended equals sign) should be used.

Block-IV Details: Cross-Section Mixing

A short summary of the primary mixing arrays, MATLS and ASSIGN, is given here for quick reference. Normally, THESE TWO ARRAYS ARE REQUIRED and, in most problems, would be the only arrays in this block. Other mixing arrays are also briefly described.

There are actually several nested levels of mixing. Each level has the job of calculating values from expressions of the form: $\Sigma_g = \sum_{i=1}^{k} N_i \sigma_{i,g}$ for each group, g. The user's job is to input the N_i for all the k components of the mixture and to specify each component, i. Component i has the cross section, $\sigma_{i,g}$. In common usage, for the first level of mixing, $\sigma_{i,g}$ is the effective microscopic cross section and N_i is the atom density of isotope i, and Σ_g is then the macroscopic cross section of some material. In a higher level of mixing, these materials may be homogenized into a single material by using their volume fractions for the N_i. With several nested levels, the user has a great deal of flexibility in defining what Σ_g is for that level. A more complete discussion of mixing will be found in the chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1.

A discussion of cross-section processing is outside the scope of this document, but it should be noted that the user needs to be aware of the processing that is inherent in the input library. For instance, for materials in which there are isotopes with cross-section resonances, self shielding of the cross sections for these isotopes may be important and this effect must have been considered in the preparation of the "effective" microscopic cross sections for these isotopes. Since the self shielding is dependent on the amounts and types of the other isotopes in the material, the "effective" cross section is strictly valid only for use in a mixture which has the same composition as was used in the self shielding calculation. If the user desires to use this same "effective" microscopic cross section in some other composition (mix) of material, it is up to the user to verify the accuracy of this approach.

Primary Mixing Arrays {Required}

Name	Description
MATLS[a] [-;MT]	Instructions for mixing "isotopes" or premixes into "materials." See details below.
ASSIGN[b] [-;NZONE]	Assignments of materials to geometric zones. See below.
PREMIX [-;-] {optional}	Instructions for mixing "isotopes" into premixes. See below.

a. The information entered in the MATLS array is written to the CCCC standard interface files NDXSRF and ZNATDN.

b. Information entered in the ASSIGN array is written to the code-dependent interface file ASGMAT.

In order to understand how cross sections are mixed and the resultant material placed in the problem, we first need a little conceptual information.

The key entities used in specifying the cross-section spatial distribution are coarse mesh, zone, isotope, and material.

The basic geometry of the problem is defined with the coarse meshes specified in Block-II. The geometric areas called zones are also defined there using the ZONES array; the ZONES array designates the zone number assigned to each coarse mesh.

Here in Block-IV, we mix cross sections and assign them to the zones created in Block-II. For the purposes of this discussion, the cross sections found on the input library belong, by definition, to "isotopes", no matter what their true nature. These "isotopes" may then be mixed to form materials, using the MATLS array. Materials are then assigned to zones using the ASSIGN array.

MATLS input array

The general form of a MATLS mix instruction is shown below:

$$\text{MATLS=}\ \ mat_1\ \ comp_1\ den_1,\ \ comp_2\ den_2,\ \text{...etc....}\ ;$$

where mat_1 is the desired character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional materials (i.e., mat_2, mat_3, and so on up to the required number, MT) are defined in subsequent strings. Each string may contain as many components as necessary (actual limit = 500). A component is usually an isotope from the library, but may also be a temporary material created by the PREMIX array (see below).

When the component is an isotope, the den_i is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 2-47).

Short form: MATLS= *ISOS*

This form specifies that there should be as many materials as isotopes and that isotope number 1 is to be used for material number 1, isotope number 2 is to be used for material number 2, and so on.

In the special case where there is only a single component in a material and its density is unity, the density entry may be omitted as in the first material below:

MATLS= mat_1 $comp_1$; mat_2 $comp_2$ den_2; ...etc.... ;

ASSIGN input array

The general form of the ASSIGN instruction is shown below:

ASSIGN= $zone_1$ mat_1 vol_1, mat_2 vol_2, ...etc.... ;

where $zone_1$ is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). mat_1, mat_2, and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" vol_1, vol_2, and so on. Additional zones (i.e., $zone_2$, $zone_3$, and so on up to the required number, NZONE) are defined in subsequent strings. Although it is highly recommended that you use character names, here it is convenient to use the numeral for the zone name because it is the same numeral entered in the ZONES array.

Short form: ASSIGN= *MATLS*

This form specifies that there are as many zones as there are materials, and that material number 1 is to be assigned to zone number 1, material number 2 to zone number 2, and so on.

NOTE: The short form ASSIGN=*MATLS* can not be used if you intend to use the ASGMOD input array described later in this section.

PREMIX input array

The PREMIX array forms temporary materials in a way exactly analogous to the way that permanent materials are formed in the MATLS array. The difference in treatment is that the temporary materials created by PREMIX exist only long enough to complete the mixing; they are not available for assignment to geometric zones, nor are they available for use in material edits.

The general form of a PREMIX mix instruction is shown below:

PREMIX= $tmat_1$ $comp_1$ den_1, $comp_2$ den_2, ...etc.... ;

where $tmat_1$ is the character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional temporary materials (i.e., $tmat_2$, $tmat_3$, and so on) may be defined in subsequent strings. A component may be either an isotope from the library or another temporary material created by PREMIX.

The PREMIX array is useful for organizing the mixing input. For instance, it is frequently useful to mix the cross sections for a molecule of water and then in subsequent mix instructions, to input the molecular density of water as opposed to entering the atom density for both hydrogen and oxygen. Other examples are to form average cross sections for an element composed of many isotopes, or to form full density materials and then in later mix instructions to put in the volume fraction of the full density material.

Character Names vs. Numeric Names

In the foregoing discussion, isotopes, materials, and zones were identified by their character names. Optionally, they may be referred to by their ordinal number. Thus, 2 for an isotope name would call for the second isotope on the library. However, this practice is NOT recommended.

THE CHARACTER NAME FORM IS HIGHLY RECOMMENDED. It provides the most straightforward, most readable form. If the character name form is used, the naming input arrays in the following table are not needed.

Using the character name form in one array and the numeric name form in another array is particularly discouraged. However, should one wish to use the numeric form in the MATLS and/or ASSIGN arrays, and then subsequently associate character names with the ordinal numbers, one can use the naming arrays in the following table to do so. This situation could arise if, for some reason, one wanted to use material numbers in the MATLS array, but use character material names in the ASSIGN array.

When the library is of the MENDF form, the character names that must be used for the isotope names are discussed in "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.

Mixing Array for a Concentration Search {Optional}

Name	Description
ASGMOD[a] [-;-]	C_1 parameters used in concentration searches. See the discussion below.

a. The information entered in the ASGMOD array is written to the ASGMAT file together with the information from the ASSIGN and CMOD arrays.

ASGMOD input array

The ASGMOD array is used in conjunction with the ASSIGN array when one wishes to vary the composition of a zone or zones in order to achieve a certain value of k-effective or alpha (i.e., in a concentration search). The concentration (or volume fraction) of material x in zone z is given by the following expression:

$$C(z,x) = C_0(z,x) + C_1(z,x)*CMOD$$

where $C_0(z,x)$ is the base concentration of material x in zone z. This is the concentration (or volume fraction) entered in the ASSIGN array for material x. In these arrays, x is not any kind of an index; correspondence is made by name, rather than by position within the array. Thus, for instance, in a problem that had ten materials, we might only assign one of them to a given zone. It would then probably be in the first position in the ASSIGN array string for that zone even though it might have been, say, sixth in the list of all materials.

$C_1(z,x)$ is the corresponding entry in the ASGMOD array for material x in zone z.

CMOD is the search parameter (sometimes called search eigenvalue) that will be varied by ONEDANT in order to achieve the desired k-effective or alpha value. In a search calculation, the initial value for CMOD will be the input value EV.

The general form of the ASGMOD instruction is shown below:

ASGMOD= *zone* mat_m vol_m, mat_n vol_n, ...etc.... ;

where *zone* is the character name of any zone in the problem, mat_m, mat_n, and so on are the character names of any of the materials that will be present in this zone, and vol_m, vol_n, and so on are the C_1 values for respectively, mat_m, mat_n, and so on. Additional zones may be specified in subsequent strings. All zones do not have to appear in the ASGMOD array nor do all materials within a zone have to appear in the string for that zone.

Concentration Modifier {Optional}

Name	Description
CMOD	Concentration modifier. Input value is not used in a search. See the discussion below.

The concentration modifier, CMOD, is varied by ONEDANT during a search calculation. For any other type of calculation, a value of CMOD may be input and the composition of the zones will be calculated using the expression above for C(z,x).

Miscellaneous Mixing Input {Optional}

Name	Comments
MATNAM [MT]	Character material names for Materials. Used only if the mat_1 name used in the MATLS array was integer. First entry in MATNAM array is the desired character name for Material number 1, second entry is the desired character name for Material number 2, etc.
ZONNAM [NZONE]	Character zone names for Zones. Used only if the zone name entry in the ASSIGN or ASGMOD array was integer. First entry in the ZONNAM array is the desired character name for Zone number 1, second entry is the desired character name for Zone number 2, etc.
MATSPEC [≤MT]	Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions. Allowable entries are the words: *ATDENS* (default) atomic densities *ATFRAC*[a] atomic fractions *WTFRAC* weight fractions Can be input as a vector with up to MT entries (one for each Material) [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13.] If less than MT entries are made, the last entry will be used to fill out the array to a length of MT.
ATWT [≤2*NISO] {required[b]}	Atomic weights of the isotopes. If using MATSPEC=ATFRAC or WTFRAC, atomic weights must be available to the code. Entries for the ATWT array are made in pairs, as follows: ATWT= iso_1 $atwt_1$ iso_2 $atwt_2$ .... where iso_n is the isotope name (identifier) for isotope n on the cross-section library and $atwt_n$ is that isotope's atomic weight. [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13].

a. ATFRAC and WTFRAC cannot be used with PREMIX.

b. Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.

Block-V Details: Solver Input

Desired Calculation

Name	Comments
IEVT	Calculation type: Enter one of the following values: Value — Calculation Desired *-1* — inhomogeneous source with fission or upscatter. *0* — inhomogeneous source alone(default). *1* — k_{eff}. *2* — α (time absorption) search. *3* — concentration search. *4* — dimension search.
ISCT	Legendre order of scattering (default=0).
ITH	Calculation Mode. *0/1* = direct/adjoint calculation (default=0).
IBL	Left boundary condition[a]. *0/1/2/3* = vacuum/reflective[b]/periodic/white[c] (default=vacuum).
IBR	Right boundary condition: Same input values as for IBL.

a. The left boundary condition applies only for planar geometry.

b. May be used in conjunction with an albedo. See the LBEDO array page 2-57.

c. May be used in conjunction with an albedo. See the LBEDO array page 2-57.

Iteration Controls

Name	Comments
EPSI	Inner iteration convergence precision (default=0.0001).
EPSO	Outer iteration convergence precision (default=EPSI).
IITL	Maximum number of inner iterations per group until $\|1.0-\lambda\|<3*EPSO$. Recommend default be used. (default is chosen by code).
IITM	Maximum number of inners per group allowed after $\|1.0-\lambda\|<3*EPSO$. Recommend default be used. (default is chosen by code).
OITM	Maximum number of outer iterations (default=20).
ITLIM	Number of seconds time limit (default=unlimited).

Acceleration Controls

Name	Comments
GREYACC	Upscatter acceleration[a]. *0/1 = no/yes*[b].

a. See page 7-21 for details.

b. The user may also input the words *no* or *yes*.

K-Code Convergence {Optional}

Name	Comments
KCALC	Special Criticality Convergence Scheme. *0/1* = no/yes.

A special convergence scheme may be invoked for problems which require a good eigenvalue but do not require tight convergence of the pointwise fluxes. It consists of converging the eigenvalue but not the pointwise fluxes. Normally both must be converged. It also sets the default for eigenvalue convergence to 0.001 rather than 0.0001. To invoke this option to save running time, set the input parameter KCALC to unity.

Output Controls {Optional}

Name	Comments
FLUXP	Final flux print. *0/1/2* = no/isotropic/all moments.
XSECTP	Cross-section print. *0/1/2* = no/principal/all .
FISSRP	Fission source rate print. *0/1* = no/yes.
SOURCP	Source print. *0/1/2/3* = no/as input/normalized/both .
GEOMP	Fine mesh geometry print. *0/1* = no/yes.
ANGP	Print angular fluxes. *0/1* = no/yes.
BALP	Print balances for each coarse mesh interval. *0/1* = no/yes.
RAFLUX	Prepare angular flux file (RAFLUX/AAFLUX). *0/1* = no/yes.
RMFLUX	Prepare flux moments file (RMFLUX/AMFLUX). *0/1* = no/yes.

Miscellaneous Solver Input {Optional}

Name	Comments
TRCOR	Apply transport correction[a] to cross sections on MACRXS file. Enter one of the following words: Word — Description *DIAG* — Use diagonal transport correction. *BHS* — Use Bell-Hansen-Sandmeier correction. *CESARO* — Use Cesaro "correction". *NO* — (or omit entry) Use no correction.
BHGT	Buckling height (in cm. if macroscopic cross section in cm^{-1}.) Used only for plane, cylindrical, and two-angle plane geometries. (default = 0.0, which is treated as infinity).
BWTH	Buckling width. Used only for plane and two-angle plane geometries. (default = 0.0, which is treated as infinity).
NORM	Normalization constant. Normalize the fission source rate to this value when IEVT≥1 or normalize the inhomogeneous source rate to this value when IEVT<1. NORM=0 means no normalization. (Integral of source rate over all angle, space, and energy = NORM, except for k_{eff} problems where the integral is equal to NORM*k_{eff}.) Any fluxes printed by setting FLUXP nonzero will be normalized consistently with this source rate.
I2ANG	Do two-angle plane calculation? *0/1* = no/yes. For IGEOM=1 only.
CHI [NGROUP;M]	Fission fraction born into each group.[b] Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.
DEN [IT]	Density factor to use in each fine mesh interval. Applied to the zone macroscopic cross sections at each mesh interval.

a. For more information, see "Transport Corrections for the Cross Sections (TRCOR)" on page 7-31.

b. This input will override any previous CHI from earlier blocks or from any cross-section library which contains CHI.

Quadrature Details

Name	Description
IQUAD	Source of quadrature constants. Enter one of the following: Value — Description *-3* — Get constants from SNCONS file. *1* — Traditional built-in P_n constants [S_2, S_4, S_6, S_8, S_{12}, S_{16}, S_{20}, S_{24}, S_{32}, or S_{48}]. (This is the default). *2* — Traditional built-in DP_n constants [S_4, S_8, S_{12}, S_{16}, S_{24}, S_{32}, S_{40}, S_{48}, S_{64} or S_{96}]. *4* — Built in GQ_n set (recommended for cylinders) [S_2, S_4, S_6, S_8, S_{12}, or S_{16}].
WGT[a] [MM[b]] {optional}	Quadrature weights.
MU [MM] {optional}	Mu cosines.

a. Presence of the WGT and MU arrays overrides the IQUAD input

b. MM=ISN for plane or spherical geometry. MM=ISN*(ISN+2)/4 for cylindrical geometry. MM=ISN*(ISN+2) for 2-angle plane geometry. For ordering of weights, see "Discrete-Ordinates Equations in One Dimension" on page 12-27. MM is the total number of angles in the problem for ONEDANT. There is a different definition of MM for the higher dimension solvers.

Flux Guess From RTFLUX {Optional}

Name	Comments
INFLUX	Read initial flux guess from the RTFLUX file. *0/1* = no/yes.

General Eigenvalue Search Control[a] { IEVT >1}

Name	Comments
IPVT	Type of eigenvalue to search for in a concentration or dimension search. *0/1/2* = none / k_{eff} / α. (default = 1).
PV	Value of k_{eff} or α to which to search. (default = 1.0 if IPVT=1, 0.0 if IPVT=2.
EV	Initial search parameter. Value at which to start the search parameter. (default=0).
EVM	Initial search parameter increment. Amount by which to change search parameter in the first step of a search. (REQUIRED - there is no default).
XLAL	Lambda lower limit for search. (default = 0.01).
XLAH	Lambda upper limit for search. (default = 0.5).
XLAX	Lambda convergence criterion for second and subsequent search steps. (default = 10*EPSI).
POD	Parameter oscillation damper. (default=1.0).

a. See "Eigenvalue Searches" on page 7-33 for definitions of these quantities.

ONEDANT can vary the composition of a zone (or zones) or the coarse mesh boundaries in order to achieve a desired k_{eff} or α value. The search input consists of the above general search control input plus input specific to the type of search being performed.

Dimension Search Input {Required if IEVT=4}

Name	Comments
RM [IM]	Radius modifiers for each coarse mesh interval, for use with dimension searches only.

The dimension search requires the RM input as well as the general search input above. During the search, ONEDANT varies the search parameter (sometimes called the search eigenvalue) denoted by EV in the following expression to change the coarse mesh boundaries:

$$XMESH_{i+1} = XMESH_i + \{XMESH_{i+1}-XMESH_i\} * [1.0+EV*RM_i], \quad i=1, \ldots, IM$$

Although it may seem a bit awkward at first, the user will find this expression to be quite flexible. With proper choice of the RM_i values, the user can move any or all of the coarse mesh boundaries while allowing others to remain stationary. The quantities in { } in the above expressions are always formed from the original input values.

Concentration Search Input {Required if IEVT=3}

Name	Description
The solver input for a concentration search is to set IEVT = 3 (page 2-48) and input the general eigenvalue search controls. But you must also input the ASGMOD[a] array in Block-IV.	

a. A concentration search involves the mixing instructions. A discussion of the ASGMOD array is found in the mixing input description on page 2-45.

Volumetric Source Options {Optional}

Name	Comments
INSORS	Read source from interface file FIXSRC. *0/1* = no/yes.
----- For a text-input source, choose one of the following options:	
Option 1:	
SOURCE [NGROUP; NMQ[a]]	Source spectrum for each of NMQ moments. (Spatial distribution is assumed to be flat with value unity)
Option 2:	
SOURCX [IT;NMQ]	X (or R) spatial distribution for each moment. (Spectrum is assumed to be flat with value unity)
Option 3: (input both arrays)[b]	
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCX [IT;NMQ]	X (or R) spatial distribution for each moment.
Option 4:	
SOURCF [IT;NGROUP*NMQ]	Spatial distribution for each row, group, and moment.

a. NMQ is not an input value but is computed from the number of strings read. NMQ must correspond exactly to the number of moments in some P_n expansion of the source. The number of moments is n+1 for plane and sphere, $(n+2)^2/4$ for n even and (n+1)(n+3)/4 for n odd for cylinder, and $(n+1)^2$ for 2 angle plane. n must be less than or equal to ISCT. See page 12-24 for more details.

b. The full source at mesh point i in group g for moment m is calculated as follows:

SOURCF(i,g,m) = SOURCE(g,m)*SOURCX(i,m)

Boundary Source Input {Optional}

Name	Comments
----- For a text-input source, choose one of the following options:	
Option 1: Isotropic Boundary Source.	
SILEFT [NGROUP]	Isotropic source on the left side in each energy group. For plane geometry only!
SIRITE [NGROUP]	Isotropic source on the right side in each energy group.
Option 2: Anisotropic Boundary Source.	
SALEFT [MM/2[a];NGROUP]	Angular flux on the left for each inwardly directed angle and group. For plane geometry only! Entered as NGROUP strings of data, each string containing MM/2 data entries, beginning with group 1. The ordering of the angular boundary sources (fluxes) is described in "Surface (Boundary) Source Input" on page 7-28.
SARITE [MM/2;NGROUP]	Angular fluxes on the right side.

a. MM = ISN for standard plane and spherical geometries (IGEOM=1,3)
MM = ISN*(ISN+2) for two-angle plane geometry (IGEOM=1 and I2ANG=1)
MM = ISN*(ISN+2)/4 for cylindrical geometry (IGEOM=2)

Albedos
{Optional}

Name		Comments
LBEDO	[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.
RBEDO	[NGROUP]	Right boundary albedoes for each group for all geometries. Applied as albedoes for either reflective (IBR=1) or white (IBR=3) boundary conditions.

Block-VI Details: Edit Input*

Spatial Specifications for Reaction Rates {Required[a]}

Name	Comments
PTED	Do edits by fine mesh. *0/1* = no/yes.
ZNED	Do edits by zone. *0/1* = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below.)
POINTS [≤IT] {optional}	Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (default= all mesh intervals)
EDZONE [IT] {optional}	Edit zone number for each fine mesh interval. USED ONLY IF ZNED=1. (default = SOLVER coarse mesh interval numbers, see ZONES array, Block-II on page 2-34).

a. Either PTED or ZNED or both must be unity in order to produce reaction rate edits.

* More details for the input for edits are given in chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

Reaction Rates from Cross Sections[a] {Optional[b]}

Name	Comments
EDXS [≤NEDT] {required[c]}	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 the character name of the cross-section type. See the table "Edit Cross-Section Types by Position and Name" on page 2-60 or "MENDF Library Edit Cross Sections" on page 2-66 for the available names. NEDT is the total number of edit cross-section types available from the input cross-section library. (default = all shown in the table) Note: The cross-section types specified in this array apply to any or all of the following edit forms: RESDNT, EDISOS, EDCONS, EDMATS.
RESDNT	Do edits using the resident macroscopic cross section at each point. *0/1* = no/yes.
EDISOS [≤NISO]	Character names of the isotopes to be used in forming Isotopic reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDCONS [≤NISO]	Character names of the isotopes to be used in forming resident constituent (partial macroscopic) reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDMATS [≤MT]	Character names of materials to be used in forming Material (macroscopic) reaction rates. The ordinal number may alternately be used, but is not recommended. (default = none).
XDF[d] [IT]	Fine mesh density factors for the x (or r) directions. The density factor is used to multiply resident constituent (see EDCONS), macroscopic (see EDMATS), and resident macroscopic (see RESDNT) reaction rates only. (default= all values unity).

a. See chapter "RUNNING THE EDIT MODULE" starting on page 8-1 for further discussion.

b. But either something in this grouping or the next must be input in order to produce reaction rate edits.

c. You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.

d. If density factors were used in SOLVER to modify the cross sections at each mesh interval, the same density factors should be entered here in the XDF array as well.

Edit Cross-Section Types by Position and Name

CROSS-SECTION INPUT VIA ISOTXS or GRUPXS			CROSS-SECTION INPUT VIA ASCII TEXT		
Type	EDIT Position	Name[a]	Type	EDIT Position	Name
chi	1	CHI.....	not used	1	CHI.....
nu-fission	2	NUSIGF..	nu-fission	2	NUSIGF..
total	3	TOTAL...	total	3	TOTAL...
absorption	4	ABS.....	absorption	4	ABS.....
(n,p)	5	N-PROT..	1[b]	5	EDIT1...[c]
(n,d)	6	N-DEUT..	2	6	EDIT2...
(n,t)	7	N-TRIT..	3	7	EDIT3...
(n,alpha)	8	N-ALPH..	.	.	.
(n,2n)	9	N-2N....	.	.	.
(n,gamma)	10	N-GAMM..	.	.	.
fission	11	N-FISS..	N=IHT-3	4+N	EDITN...
transport	12	TRNSPT..			

a. Names are eight characters. A period within a name in this table denotes a blan.k

b. Denotes position in the cross-section table. All cross sections in 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 EDNAME array of Block-III.

Reaction Rates from User Response Functions {Optional[a]}

Name	Comments
RSFE [NGROUP;M] {required}	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 with NGROUP entries beginning with group 1.
RSFX [IT;M][b] {optional}	Response function X(or R) distribution for M functions. Entered as M strings of IT entries beginning with mesh point 1. (default=1.0 in all positions).
RSFNAM [M] {optional}	Character names for the M user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM).

a. But either something from this grouping or the previous one must be input in order to produce reaction rate edits.

b. The m-th response function at space point i and energy group g is computed as:

RSFX(i,m) * RSFE(g,m)

Energy Group Collapse Specifications {Optional}

Name	Comments
ICOLL [NBG]	Edit energy group collapsing option: Number of SOLVER energy groups in each EDIT broad group. The NBG entries must sum to NGROUP. (default = 1 energy group per EDIT broad group)
IGRPED	Print option on energy groups. Enter one of the following values:

Value	Description
0	Print energy group totals only
1	Print broad groups only
2	Print broad groups only (same as 1)
3	Print both broad groups and totals

Reaction Rate Summing {Optional}

Name	Comments
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 "Edit Cross-Section Types by Position and Name" on page 2-60). Each of 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. (for more detail, see "Response Function Summing Options" on page 8-13.)
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 page 8-13 for more detail.

Mass Inventories {Optional}

Name	Comments
MASSED	Calculate and print mass inventories by zone. *0/1/2/3* = none/ solver zones/edit zones/both (default=1). This option is active only if atomic weights are present. See ATWT on page 2-47.

Power Normalization {Optional}

Name	Comments
POWER {required}	Normalize to POWER megawatts.[a] All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be normalized. Fluxes are normally not printed here in the EDIT module, although they may be extracted by using a unit response function. Any such fluxes will also be normalized to POWER. Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 2-51).
MEVPER {required}	MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power. For the power calculation, ONEDANT needs to know which cross section is the fission cross section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or GRUPXS library that designation is automatically provided (See "Edit Cross-Section Types by Position and Name" on page 2-60). But if one uses an ASCII text library, either ODNINP or XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 2-39).

a. Note that this normalization is meaningless if you are using the results of an adjoint run.

Miscellaneous Edit Items {Optional}

Name	Comments
RZFLUX	Write the CCCC standard zone[a] flux file RZFLUX or AZFLUX. *0/1* = no/yes.
RZMFLX	Write the code-dependent zone[b] flux moments file RZMFLX or AZMFLX. *0/1* = no/yes.
EDOUTF[c]	ASCII output files control. Enter one of the following values: Value — Description *-3* — Write both EDTOGX (without scalar fluxes) and EDTOUT files. *-2* — Write EDTOGX file (without scalar fluxes). *0* — Write neither file. (default) *1* — Write EDTOUT file. *2* — Write EDTOGX file (with scalar fluxes). *3* — Write both EDTOGX (with scalar fluxes) and EDTOUT files.
BYVOLP	Printed point reaction rates will have been multiplied by the mesh volume. *0/1* = no/yes.
AJED[d]	Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. *0/1* = regular/adjoint.
FLUXONE	Flux override. *0/1* = no/yes. Replaces all the input fluxes by unity. Useful for seeing the cross sections used in cross-section edits. WARNING! Meaningful reaction rates cannot be obtained when this switch is on.

a. RZFLUX and AXFLUX are organized by solver zones.

b. RZMFLX and AZMFLX are organized by solver zones.

c. See "ASCII File Output Capabilities (the EDOUTF Parameter)" on page 8-15.

d. See "Adjoint Edits" on page 8-15.

MENDF Library Edit Cross Sections

Reaction Type	Name	Description
χ	CHI	fission spectrum
$\nu\sigma_f$	NUSIGF	effective nu-sigma-fission
σ_t	TOTAL	Total cross section
σ_a	ABS	absorption [a]
(n,n)	MEND1	elastic scattering
(n,n′)	MEND2	inelastic scattering
(n,2n)	MEND3	n,2n scattering
(n,3n)	MEND4	n,3n scattering
(n,γ)	MEND5	gamma production
(n,α)	MEND6	alpha production
(n,p)	MEND7	proton production
(n,f)	MEND8	direct fission
(n,n′)f	MEND9	second-chance fission
(n,2n)f	MEND10	third-chance fission
(n,F)	N-FISS	[(n,F) = (n,f) + (n,n′)f + (n,2n)f]
χ_p	MEND12	prompt fission spectrum (only for fissionable materials)
χ_t	MEND13	total fission spectrum (only for fissionable materials)

a. σ_a for group g is defined as $\sigma_a = \sigma_t - \sum_{g'} \sigma_{g \to g'}$.

When using the Los Alamos MENDF5 cross-section library with the codes, there are numerous edit cross sections available for use in the Edit Module. Since these come from the MENDF file, they are called upon with special character names in the Edit Module as part of the EDXS input. These names are defined in the table above.

REFERENCES

1. 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.
2. 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.
3. 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).
4. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," *Nucl. Sci. Eng.* **64**, 344 (1977).
5. R. D. O'Dell, F. W. Brinkley, D. R. Marr, and R. E. Alcouffe, "Revised User's Manual for ONEDANT: A Code Package for One- Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory report LA-9184-M, Revised, (December 1989).
6. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).
7. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693, (March 1967).

APPENDIX A: SAMPLE INPUT

This appendix presents the printed output from each of 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.

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 (page 2-74) 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 on page 2-20.

On page 2-75 of the problem output is a descriptive summary of the Title Card Control Parameters and a 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 2-76 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 the tables "Edit Cross-Section Types by Position and Name" on page 2-60 and "MENDF Library Edit Cross Sections" on page 2-66.) 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. Character names have been provided through the NAMES array in Block-III, and these are listed under the column labeled Isotope Name. The scattering is specified to be isotropic, and this is indicated by the entries "p0" under the column labeled 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.)

On page 2-77 of the output the user is provided 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 the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure" starting on page 13-1. 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.

On page 2-78 the printed output from by the Solver Module begins. There is first presented 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 labeled RAW INPUT and one labeled 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, 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 of EPSI (0.0001) is, accordingly, assumed by the Solver Module and this value is provided under the AS DEFAULTED column.

On page 2-79 of the output are more of the Solver input parameters and also listed are 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.

On page 2-80 of the output is provided a recap of the assignment of materials to zones in terms of the algorithm described on page 2-45 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 Figure 12.2 in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. The column labeled LI refers to the ξ-level index. The terms BETA PLUS and BETA MINUS refer, respectively to the terms

$\alpha_{m+1/2}/w_m$ and $\alpha_{m-1/2}/w_m$ in Eq. (25) on page 12-30. (For spherical geometry BETA PLUS and BETA MINUS refer, respectively, to one-half the value of the terms $\beta_{m+1/2}/w_m$ and $\beta_{m-1/2}/w_m$ in Eq. (28) on page 12-31

The next page of the output (page 2-81) 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), $\nu\sigma_f$, σ_t, and σ_a.* The scattering matrix terms correspond to the coefficients in a Legendre expansion of the term $\sigma_{s,\,g'\rightarrow g}(r, \mu_0)$ in Eq. (2) on page 12-11. The value of the Legendre order for the term is provided under the column labeled 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

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

The entries in the column labeled FIRST GRP 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 the output page is geometry and spatial mesh information.

On page 2-82 is provided a summary description of iteration control criteria followed by the iteration monitor print. These items are fully described on page 7-19. 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_{eff} = 0.993 402 is provided in the monitor print for outer iteration 4 under the column labeled K-EFF EIGENVALUE.

Then page 2-83 provides a balance table print for each energy group and the sum of the groups. The group-dependent quantities are defined and computed as follows:

(1) SOURCE = total inhomogeneous source = QG_g

$$QG_g = \sum_{i=1} Q_i V_i + \sum_{\mu_m<0} w_m|\mu_m|A_{IT+1/2}QR_m + \sum_{\mu_m>0} w_m\mu_m A_{1/2}QL_m\ ,$$

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

* Note that in this discussion, as in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1, a lower case sigma is used to represent a macroscopic cross section.

interval i, $A_{IT+1/2}$ is the surface area at the rightmost boundary of the system, and $A_{1/2}$ is the surface area at the leftmost boundary of the system;

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

For a k-effective eigenvalue problem,

$$FG_g = \frac{1}{k_{eff}} \sum_{h=1}^{NGROUP} \sum_{i=1}^{IT} \chi_{g,i} (\nu\sigma_f)_{h,i} \phi_{h,i} V_i ;$$

For a source with fission problem,

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

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

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

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

$$SS_g = \sum_{i=1}^{IT} \left(\sigma^o_{s,g \rightarrow g} \right)_i \phi_{g,i} V_i ;$$

(5) OUT SCATTER = out-scatter from group g to all other groups = $SOUT_g$

$$SOUT_g = \sum_{i=1}^{IT} \left(\sigma'_{t,g} \right)_i \phi_{g,i} V_i - ABG_g - SS_g ;$$

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

(6) ABSORPTION = absorption in group g = ABG_g

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

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

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

$$RL_g = \sum_{\mu_m > 0} w_m \mu_m A_{IT+1/2} \psi_{m,IT+1/2} - \sum_{\mu_m < 0} w_m |\mu_m| A_{IT+1/2} \psi_{m,IT+1/2} \ ;$$

(8) NET LEAKAGE = net flow from system (both boundaries) = NL_g

$$NL_g = RL_g + \sum_{\mu_m < 0} w_m |\mu_m| A_{1/2} \psi_{m,1/2} - \sum_{\mu_m > 0} w_m \mu_m A_{1/2} \psi_{m,1/2} \ ;$$

(9) PARTICLE BALANCE = BAL_g

$$BAL_g = 1 - \frac{NL_g + ABG_g + SOUT_g}{QG_g + FG_g + SIN_g};$$

(10) NPROD = spectrum of neutrons causing fissions = $NPROD_g$

$$NPROD_g = \frac{1}{N} \sum_{i=1}^{IT} (\nu\sigma_f)_{g,i} \phi_{g,i} V_i \ ;$$

$$\text{where - } N = \sum_{g=1}^{NGROUP} \sum_{i=1}^{IT} (\nu\sigma_f)_{g,i} \phi_{g,i} V_i \ .$$

On page 2-84 are provided 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 produced by neutrons in energy group g at mesh point i is simply the quantity $(\nu\sigma_f)_{g,i} \phi_{g,i}$, having units of particles per unit time and volume.

The final page (page 2-85), provides the RUN HIGHLIGHTS for the sample problem execution. This is followed by a storage and timing history of the run.

It is to be noted that no Edit Module output appears in the output of this sample problem. The reason for this is that there is no edit input available. 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  8/25/94 with solver version 08-04-94beta---- release 2.4m     machine tumbler

                    ...listing of cards in the input stream...

  1.    2    0    0
  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= cdnimp
 26.     maxord=0  ihm=6  iht=4  ihs=5  ifido=0  ititl=1
 27.     names= "o-16"  "na-23"  fe  cr  ni  "pu-239"  "u-238"
 28.     cdname= fiss
 29.  /
 30.  / ****** since lib=cdnimp, 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  (o-16)    sample 2 group lmfbr cross sections
 37.        0.000       0.010       0.000       2.000       1.600       0.000 o16/1
 38.        0.000       0.000       0.000       3.600       3.600       0.390 o16/2
 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 group 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, fe .05, cr .016,  ni 0.01;
 65.            fuel  "pu-239" .0051,  "u-238" .0155    "o-16" .0412;
 66.            blkt  "u-238" 0.0206,  "o-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  niso    number of input isotopes (from isotxs, 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 intervals
*
*                                                 ...storage...
*
*                                               mexlcm=  140000
*                                               mexscm=   40000
*
*********************************************************************************************************************************
******************************
*key end   block ii read-geom*
******************************
*
*                         2  igeom    1/2/3/6/7/8/9/11/14
******************************
*key end   block iii read-xs *
******************************
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                        ...block iii - cross-section library...
*
**********************************************************************************************************************************
*
*                                                ...library source...
*                                                   lib=cdnlmp
*
*                                             ...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   -1/0/1/2 - dtf(4e18.0)/dtf/fixed fido/free fido library
*                        1  ititl   0/1 - no/yes there is a title card before each table
*                        0  i2lpl   0/1 - no/yes library higher order scattering contains 2l+1 factor
*                        0  savxs   0/1 - no/yes save binary xslib file (filename=bxslib)
*                        1  kwikrd  0/1 - full fido read/quick fido read (default=quick)
*
*
*                                                ...energy structure...
*
*   group      chi          vel        lower bound   upper bound   group      chi          vel        lower bound   upper bound
*   -----  -----------  -----------  -----------   -----------   -----  -----------  -----------  -----------   -----------
*     1    0.00000E+00  0.00000E+00  0.00000E+00   0.00000E+00       2  0.00000E+00  0.00000E+00  0.00000E+00   0.00000E+00
*
*                                      last neutron group(lng) is number   2
*
*
*                        0  balxs   -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance
*
*
*                                                ...edit position names...
*
*                                                                    card
*                                                position  edname  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 isotope
*                                 number  name   order          header card
*                                 ------  -----  -----  ------------------------------
*                                    1.   o-16    p0   -  oxygen-16  (o-16)   sample 2 group lmfbr cross sections
*                                    2.   na-23   p0   -  sodium     (na-23)  sample 2 group lmfbr cross sections
*                                    3.   fe      p0   -  iron       (fe)     sample 2 group lmfbr cross sections
*                                    4.   cr      p0   -  chromium   (cr)     sample 2 group lmfbr cross sections
*                                    5.   ni      p0   -  nickel     (ni)     sample 2 group lmfbr cross sections
*                                    6.   pu-239  p0   -  plutonium  (pu-239) sample 2 group lmfbr cross sections
*                                    7.   u-238   p0   -  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. core     fuel      3.50000E-01, sodium   4.00000E-01, steel    2.50000E-01,
*       2. blankt   blkt      3.50000E-01, sodium   4.00000E-01, steel    2.50000E-01,
*       3. shield   sodium    7.00000E-01, steel    3.00000E-01,
************************
*key start mix card xs *
************************
************************
*key end   mix card xs *
************************
*****************************************************************************************************************************************
*****************************
*key end   block v read-solvr*
*****************************
************************
*key end   input module*
************************
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
```

```
*********************************************************************************************************************************************
*********************************************************************************************************************************************
*
*                  this onedant problem run on  8/25/94 with solver version 08-04-94*product release 2.4m      machine tumbler
*
*********************************************************************************************************************************************
*******************************
*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)...
*
*          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)...
*
*       0.000E+00 1.000E-04  epsi   inner iteration convergence criterion (default=0.0001)
*       0.000E+00 1.000E-04  epso   outer iteration convergence criterion (default=epsi)
*          0         1       iitl   maximum number of inner iterations per group until (1-lambda).lt.3*epso (default calculated)
*          0        20       iitm   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       kcalc  0/1, no/yes, only converge the eigenvalue (to 0.001)
*          0         0       itlim  iteration time limit (seconds)
*
*
*
*********************************************************************************************************************************************
*********************************************************************************************************************************************
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                    ...block v -- solver input (continued)...
*
*****************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*
*                                          ...miscellaneous parameters(array name = misc)...
*
*      0.000E+00 0.000E+00  bhgt   buckling height
*      0.000E+00 0.000E+00  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
*          0         1      iquad  -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  xsectp 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 inhomogeneous 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  zmflux 0/1     no/yes - write flux moments to file zmflux
*                     0  balp   0/1     no/yes - print coarse mesh balances
*
*
*                                          ...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  igan   source anisotropy
*                                   0  isorse number of source vectors input
*                                   0  isorsx number of sourcx vectors input
*                                   0  isorsf number of sourcf vectors input
*                                   0  iql    -1/0/1 isotropic/none/all angles - left boundary source
*                                   0  iqr    -1/0/1 isotropic/none/all angles - right boundary source
*                                   0  isbedo 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)*( c0 + c1*cmod )
                                  with
                               cmod =  0.000000E+00

                          entry   zone      material        c0              c1
                                no. name    no. name

                            1    1 core     2 fuel       3.500000E-01   0.000000E+00
                            2    1 core     4 sodium     4.000000E-01   0.000000E+00
                            3    1 core     1 steel      2.500000E-01   0.000000E+00
                            4    2 blankt   3 blkt       3.500000E-01   0.000000E+00
                            5    2 blankt   4 sodium     4.000000E-01   0.000000E+00
                            6    2 blankt   1 steel      2.500000E-01   0.000000E+00
                            7    3 shield   4 sodium     7.000000E-01   0.000000E+00
                            8    3 shield   1 steel      3.000000E-01   0.000000E+00


...zone assignment map...

*key start map   *

          r******************
          r     *     *     v
          r   1 *   2 *   3 v
          r     *     *     v
          r******************
     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       1324     40000
                                large core        670    140000

*key start quadrature  *

...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  0.000000E+00  0.86113631    131.9715
   2      6     2   1.739274E-01 -8.611363E-01 -1.497752E-01  8.611363E-01  0.000000E+00  0.33998104    156.3027
   3      5     2   1.630363E-01 -3.399810E-01 -5.542925E-02  1.258643E+00  9.186619E-01  0.33998104    111.1933
   4      1     1   1.630363E-01  3.399810E-01  5.542925E-02  0.000000E+00  3.399810E-01  0.86113631     48.0285
   5      3     2   1.630363E-01  3.399810E-01  5.542925E-02  9.186619E-01  1.258643E+00  0.33998104     68.8067
   6      2     2   1.739274E-01  8.611363E-01  1.497752E-01  0.000000E+00  8.611363E-01  0.33998104     23.6973


...fission guess-flux zeroed in all groups...
```

```
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                    ...cross-section related data from file macros  000000100193 version  1 ...
*
*********************************************************************************************************************************
*
*                                        ...matls available from file macros...
*
*  1 steel     2 fuel      3 blkt      4 sodium
*
*
*********************************************************************************************************************************

*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                        ...cross sections for legendre orders up to p0...
*
*********************************************************************************************************************************
*
*
*****************************
*key start mac cross sections*
*****************************
*
************************************************** g r o u p   1 ***************************************************************
*
*                                        ...principal cross sections...
*
*                          zone          chi          nu*fission       total        absorption
*                        no.  name
*                         1  core      6.0000E-01    1.6074E-02    1.2396E-01    6.1670E-03
*                         2  blankt    7.0000E-01    6.4890E-03    1.2378E-01    3.4002E-03
*                         3  shield    7.0000E-01    0.0000E+00    8.3710E-02    4.5740E-04
*
*                                        ...scattering matrices...
*                                           (2l+1 not included)
*
*   zone   order  first grp    cross sections
*     1      0      1          9.2767E-02
*     2      0      1          9.4552E-02
*     3      0      1          6.8070E-02
*
************************************************** g r o u p   2 ***************************************************************
*
*                                        ...principal cross sections...
*
*                          zone          chi          nu*fission       total        absorption
*                        no.  name
*                         1  core      4.0000E-01    8.5680E-03    2.8011E-01    7.5050E-03
*                         2  blankt    3.0000E-01    0.0000E+00    2.8189E-01    3.9350E-03
*                         3  shield    3.0000E-01    0.0000E+00    1.8550E-01    4.2350E-04
*
*                                        ...scattering matrices...
*                                           (2l+1 not included)
*
*   zone   order  first grp    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  0.00000000E+00
*                              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                                 1      terminates inners
           until (1.-lambda).lt.3*epso
   iitm  - inner iteration count                                20      terminates inners
           after (1.-lambda).lt.3*epso
   epsi  - fractional ptwise flux change                  1.00E-04      does another inner
           per inner

                                        diffusion sub-outers

criterion    quantity to test                                  value    action taken if value exceeded
   oitmd - 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                  0.95*epsx     does another sub-outer
           per 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   outer        diffusion      k-eff                          max ptwise   inners
 (sec)   no.  inners sub-outers  eigenvalue    lambda-1          flux change converged
  0.10     0
  0.23     1      0        8    9.87880E-01  2.71041E-05       6.71213E-04   yes
  0.28     2      2        3    9.93480E-01  4.57919E-03       8.75653E-02   **no**
  0.33     3      2        2    9.93410E-01 -6.25558E-05       1.63735E-02   **no**

                  — inner iteration summary for outer iteration no.   4 —

                                        iter per  max flux   at
                                   group  group    change   mesh
                                     1      3    0.57E-04    47
                                     2      2    0.87E-04    50

                $$$$$$ all convergence criteria satisfied $$$$$$

 time   outer        diffusion      k-eff                          max ptwise   inners
 (sec)   no.  inners sub-outers  eigenvalue    lambda-1          flux change converged
  0.40     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... (neutrons only)
*
*
*******************************
*key start balance tables     *
*******************************
*              gp          source      fission source          in scatter        self scatter         out scatter
*
*               1   0.0000000E+00       6.0245109E-01       1.1102230E-16       1.8227508E+00       4.9157276E-01
*               2   0.0000000E+00       3.9754891E-01       4.9157276E-01       3.9595671E+01       7.1054274E-15
*
*           total   0.0000000E+00       1.0000000E+00       4.9157276E-01       4.1418422E+01       4.9157276E-01
*
*
*              gp       absorption      particle balance      right leakage        net leakage       nprod spectrum
*
*               1   1.0871255E-01      -1.4088485E-07       2.1659566E-03       2.1659566E-03       2.7579733E-01
*               2   8.3207469E-01      -2.6216064E-06       5.7049225E-02       5.7049225E-02       7.2420267E-01
*
*           total   9.4078723E-01      -1.6196348E-06       5.9215182E-02       5.9215182E-02       1.0000000E+00
*
*
*********************************************************************************************************************************
*********************************************************************************************************************************



integral summary information
summary integral-k-eff     9.9340174E-01
integral-source-i              neutron  0.0000000E+00
integral-fission-i             neutron  1.0000000E+00
integral-in-scak-i             neutron  4.9157276E-01
integral-self-scak-i           neutron  4.1418422E+01
integral-out-scak-i            neutron  4.9157276E-01
integral-absorption-i          neutron  9.4078723E-01
integral-right lkage-i         neutron  5.9215182E-02
integral-net lkage-i           neutron  5.9215182E-02
*
*
*                                        execution time          0.47 seconds
*
*

*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                    ...group fluxes and fission source rates by spatial mesh...
*
*********************************************************************************************************************************
```

```
******************
*key start fluxes*
******************
*
*********************************************************** g r o u p  1 ***************************************************************
*
*
*                                                          ...isotropic flux component...
*
*   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   1.2665E-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   7.0603E-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   1.2246E-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
*     1    1   7.6466E-05      1    11   6.1094E-05      2    21   8.2562E-06      2    31   9.9952E-07      3    41   0.0000E+00
*     1    2   7.6231E-05      1    12   5.8174E-05      2    22   6.4350E-06      2    32   8.2185E-07      3    42   0.0000E+00
*     1    3   7.5633E-05      1    13   5.5035E-05      2    23   5.1112E-06      2    33   6.7615E-07      3    43   0.0000E+00
*     1    4   7.4745E-05      1    14   5.1682E-05      2    24   4.1076E-06      2    34   5.5651E-07      3    44   0.0000E+00
*     1    5   7.3575E-05      1    15   4.8116E-05      2    25   3.3259E-06      2    35   4.5815E-07      3    45   0.0000E+00
*     1    6   7.2133E-05      1    16   4.4332E-05      2    26   2.7062E-06      3    36   0.0000E+00      3    46   0.0000E+00
*     1    7   7.0426E-05      1    17   4.0309E-05      2    27   2.2094E-06      3    37   0.0000E+00      3    47   0.0000E+00
*     1    8   6.8461E-05      1    18   3.5999E-05      2    28   1.8081E-06      3    38   0.0000E+00      3    48   0.0000E+00
*     1    9   6.6246E-05      1    19   3.1291E-05      2    29   1.4821E-06      3    39   0.0000E+00      3    49   0.0000E+00
*     1   10   6.3787E-05      1    20   2.5954E-05      2    30   1.2165E-06      3    40   0.0000E+00      3    50   0.0000E+00
*
*********************************************************** g r o u p  2 ***************************************************************
*
*
*                                                          ...isotropic flux component...
*
*   zone  pt                  zone  pt                  zone  pt                  zone  pt                  zone  pt
*     1    1   2.4258E-02      1    11   1.9697E-02      2    21   9.8559E-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   2.3990E-02      1    13   1.7960E-02      2    23   8.0172E-03      2    33   2.6410E-03      3    43   1.0115E-03
*     1    4   2.3726E-02      1    14   1.7021E-02      2    24   7.2069E-03      2    34   2.3607E-03      3    44   8.9296E-04
*     1    5   2.3378E-02      1    15   1.6043E-02      2    25   6.4678E-03      2    35   2.1138E-03      3    45   7.7713E-04
*     1    6   2.2949E-02      1    16   1.5033E-02      2    26   5.7966E-03      3    36   1.9286E-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   2.1858E-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
*     1    1   2.0784E-04      1    11   1.6876E-04      2    21   0.0000E+00      2    31   0.0000E+00      3    41   0.0000E+00
*     1    2   2.0709E-04      1    12   1.6155E-04      2    22   0.0000E+00      2    32   0.0000E+00      3    42   0.0000E+00
*     1    3   2.0555E-04      1    13   1.5389E-04      2    23   0.0000E+00      2    33   0.0000E+00      3    43   0.0000E+00
*     1    4   2.0328E-04      1    14   1.4583E-04      2    24   0.0000E+00      2    34   0.0000E+00      3    44   0.0000E+00
*     1    5   2.0030E-04      1    15   1.3745E-04      2    25   0.0000E+00      2    35   0.0000E+00      3    45   0.0000E+00
*     1    6   1.9662E-04      1    16   1.2881E-04      2    26   0.0000E+00      3    36   0.0000E+00      3    46   0.0000E+00
*     1    7   1.9228E-04      1    17   1.1996E-04      2    27   0.0000E+00      3    37   0.0000E+00      3    47   0.0000E+00
*     1    8   1.8728E-04      1    18   1.1100E-04      2    28   0.0000E+00      3    38   0.0000E+00      3    48   0.0000E+00
*     1    9   1.8167E-04      1    19   1.0202E-04      2    29   0.0000E+00      3    39   0.0000E+00      3    49   0.0000E+00
*     1   10   1.7549E-04      1    20   9.3109E-05      2    30   0.0000E+00      3    40   0.0000E+00      3    50   0.0000E+00
*
*********************************************************** g r o u p  s u m ***********************************************************
*
*
*                                                          ...isotropic flux component...
*
*   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   2.8376E-02      1    14   2.0236E-02      2    24   7.8399E-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   3.3937E-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
*     1    1   2.8431E-04      1    11   2.2986E-04      2    21   8.2562E-06      2    31   9.9952E-07      3    41   0.0000E+00
*     1    2   2.8332E-04      1    12   2.1972E-04      2    22   6.4350E-06      2    32   8.2185E-07      3    42   0.0000E+00
*     1    3   2.8118E-04      1    13   2.0892E-04      2    23   5.1112E-06      2    33   6.7615E-07      3    43   0.0000E+00
*     1    4   2.7803E-04      1    14   1.9752E-04      2    24   4.1076E-06      2    34   5.5651E-07      3    44   0.0000E+00
*     1    5   2.7388E-04      1    15   1.8557E-04      2    25   3.3259E-06      2    35   4.5815E-07      3    45   0.0000E+00
*     1    6   2.6876E-04      1    16   1.7314E-04      2    26   2.7062E-06      3    36   0.0000E+00      3    46   0.0000E+00
*     1    7   2.6270E-04      1    17   1.6027E-04      2    27   2.2094E-06      3    37   0.0000E+00      3    47   0.0000E+00
*     1    8   2.5574E-04      1    18   1.4700E-04      2    28   1.8081E-06      3    38   0.0000E+00      3    48   0.0000E+00
*     1    9   2.4792E-04      1    19   1.3331E-04      2    29   1.4821E-06      3    39   0.0000E+00      3    49   0.0000E+00
*     1   10   2.3927E-04      1    20   1.1906E-04      2    30   1.2165E-06      3    40   0.0000E+00      3    50   0.0000E+00
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*
*                                                          ...interface file sncons written...
*
*
*                                                          ...interface file rtflux written...
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*
*                                                    run highlights
*****************************
*key start run highlights    *
*****************************
*                                          ****************************************
*                                          *                                      *
*                                          *    all modules are tentatively go.   *
*                                          *     interface file geodst written.   *
*                                          *       cross sections from cards.     *
*                                          *    interface mixing files written.   *
*                                          *     interface file asgmat written    *
*                                          *    xs files macxs,snxedt written.    *
*                                          * *left boundary condition overridden* *
*                                          *     interface file solinp written.   *
*                                          *   edit module execution suppressed.  *
*                                          *  neither editit nor edit cards exist.*
*                                          *        start solver execution.       *
*                                          *     all convergence criteria met.    *
*                                          *     interface file sncons written.   *
*                                          *     interface file rtflux written.   *
*                                          *                                      *
*                                          ****************************************
*
*
*                                                 storage and timing history
*                                          ************************************************************
*                                          *        scm      scm      lcm      lcm      cpu     sysio  *
*                                          * module words    limit    words    limit    seconds seconds *
*                                          * ------ -----    -----    -----    -----    ------- -------*
*                                          *    0       0        0        0        0      3.0     0.0   *
*                                          *  100    3706    40000        0        0      2.2     0.0   *
*                                          *  101       0        0        0        0      0.8     0.0   *
*                                          *  102    1038    40000        5   140000      0.1     0.0   *
*                                          *  103      50    40000        0        0      0.0     0.0   *
*                                          *  104    2074    40000        0        0      0.1     0.0   *
*                                          *  105       0        0        0        0      0.0     0.0   *
*                                          *  106       0        0        0        0      0.0     0.0   *
*                                          *  107     215    40000       92   140000      0.1     0.0   *
*                                          *  108     470    40000        7   140000      0.0     0.0   *
*                                          *  109       0        0        0        0      0.0     0.0   *
*                                          *  112       0        0        0        0      0.0     0.0   *
*                                          *  200    1324    40000      670   140000      0.8     0.0   *
*                                          *  201       0        0        0        0      0.2     0.0   *
*                                          *  202     261    40000        0        0      0.0     0.0   *
*                                          *  203     261    40000        0        0      0.0     0.0   *
*                                          *  204    1324    40000       39   140000      0.1     0.0   *
*                                          *  205    1299    40000        0        0      0.1     0.0   *
*                                          * 1205       0        0        0        0      0.1     0.0   *
*                                          *  206    1310    40000        0        0      0.1     0.0   *
*                                          *  207     162    40000        0        0      0.0     0.0   *
*                                          *  400       0        0        0        0      0.0     0.0   *
*                                          ************************************************************
```

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 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 "PIECEWISE EXECUTION" on page 13-19.

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.

The first page of the output (page 2-88) 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 the first page 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.

Then page 2-89 of the output lists the Edit Module input as provided in the Block-VI card-image input. The chapter "RUNNING THE EDIT MODULE" starting on page 8-1 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 on page 2-21 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 $\nu\Sigma_f$ and Σ_f cross-section types.

On page 2-90 is provided 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.

On page 2-91 is provided 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).

The final page (page 2-92) shows the RUN HIGHLIGHTS and STORAGE/TIMING HISTORY for this sample problem.

```
generalized input module run on  8/25/94 with solver version 08-04-94beta---- release 2.4m     machine tumbler

                    ...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 geodst standard interface file
    5.  /   cross sections - from previous macros and smxedt interface files
    6.  /   mixing         - from previous nobsrf and znatdn 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.      igeom=2,ngroup=2, isn=4 niso=7  mt=4 nzone=3  im=3 it=50 nosolv=1 t
   14.  /
   15.  /
   16.  / * * * * block vi (edits) * * * *
   17.      pted=1  zned=1  points=8i1,10 3i46,50   igrped=0
   18.      edxs= nusigf fiss   resdnt=1   edmats=fuel
   19.      edcons= "pu-239"   "u-238"     t

                                    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 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 igeom   1/2/3  plane/cylinder/sphere
  2 ngroup  number of energy groups
  4 isn     angular quadrature order
  7 niso    number of input isotopes (from isotxs, 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 intervals

                            ...execution controls...

  0 nofgen  0/1     - no/yes suppress reading any more card blocks and generating associated files
  1 nosolv  0/1     - no/yes suppress solver execution
  0 noedit  0/1     - no/yes suppress edit execution

                              ...storage...

                             maxlcm=  140000
                             maxscm=   40000

*key end   block vi read-edit*

*key end   input module*

edit run on  8/25/94 with solver version 08-04-94*product release 2.4m     machine tumbler
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                                ...edit output...
*
*****************************************************************************************************************************
*
*
*                                             ...block vi - edit specification data...
*
*
***********************
*key start edit output *
***********************


*************************************************************************************************
*
*
* cross-section balancing (balxs.ne.0)
*       or
* transport correction (trcor=diag, cesaro, or bhs)
* will NOT be reflected in edits
*
*
*************************************************************************************************


*                                          ...input control integers...
*
*                    1  pted      0/1     no/yes - point edits desired
*                    1  zned      0/1     no/yes - zone edits desired
*
*                    0  ajed      0/1     direct/adjoint edit (use rtflux/atflux file)
*
*                    0  igrped    0/1/2/3 print totals only/print broad groups only/same as 1/print all groups and totals
*                    0  byvolp    0/1     no/yes - multiply point reaction rates by mesh volumes
*                    0  rzflux    0/1     no/yes - write the rzflux file (zone average flux file)
*                    0  rzmflx    0/1     no/yes - write the rzmflx file (zone average flux moments file)
*
*
*                                          ...floating parameters...
*
*           0.000000E+00  power     0/p     no/normalize all results, including flux files, to p megawatts
*           2.100000E+02  mevper    mev per fission (default: 210 mev)
*
*
*
*                                       ...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 broad 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
*                                          1  0/1/2/3  no/solver/edit/both  zones mass edit
*                                                (requires atomic weights to be present)
*
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                                ...edit output...
*
*****************************************************************************************************************************
*
*
*                                       ...edit specification data(continued)...
*
*
*                                          ...description of cross-section edits...
*
*                                isotope  no.      material  no.      constituent  no.
*                                -------  ---      --------  ---      -----------  ---
*
*                                -none-            fuel       2          pu-239     6
*                                                  resdnt                u-238      7
*
*
*                                     reaction rates will be formed for each
*                                                of the above
*                                     using the cross-section types shown below
*
*                                                    position
*                                               type    no.
*                                               ----    ---
*                                               nusigf   2
*                                               fiss     5
*
*****************************************************************************************************************************
*****************************************************************************************************************************
```

```
*******************************
*key start      materials    *
*******************************

                              ****************************
                              *                          *
                              *      reaction rates      *
                              *          from            *
                              *        materials         *
                              *                          *
                              ****************************

                          ...point edit for the sum of the neutron 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   8.03380E-04   2.63234E-04
                    4      7.0000   7.94363E-04   2.60282E-04
                    5      9.0000   7.82500E-04   2.56398E-04
                    6     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   9.10987E-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   0.00000E+00   0.00000E+00
                   47     93.0000   0.00000E+00   0.00000E+00
                   48     95.0000   0.00000E+00   0.00000E+00
                   49     97.0000   0.00000E+00   0.00000E+00
                   50     99.0000   0.00000E+00   0.00000E+00

                          ...zone edit for the sum of the neutron 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   0.00000E+00   0.00000E+00
                  ---------------  ------------  ------------
                  sum 3.1416E+04   9.93402E-01   3.25867E-01
```

```
******************************
*key start   constituents   *
******************************
*                              ******************************
*                              *                            *
*                              *       reaction rates       *
*                              *           from             *
*                              *       constituents         *
*                              *                            *
*                              ******************************
*
*
*                          ...point edit for the sum of the neutron groups ...
*
*
*                             ... constituent isotope   6 - pu-239   ...
*                        point    av rad      nusigf        fiss
*                            1    1.0000   2.61083E-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   7.68717E-05
*                            9   17.0000   2.27798E-04   7.45349E-05
*                           10   19.0000   2.19900E-04   7.19541E-05
*                           46   91.0000   0.00000E+00   0.00000E+00
*                           47   93.0000   0.00000E+00   0.00000E+00
*                           48   95.0000   0.00000E+00   0.00000E+00
*                           49   97.0000   0.00000E+00   0.00000E+00
*                           50   99.0000   0.00000E+00   0.00000E+00
*
*                             ... constituent isotope   7 - u-238    ...
*                        point    av rad      nusigf        fiss
*                            1    1.0000   2.32259E-05   7.74196E-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   0.00000E+00   0.00000E+00
*                           47   93.0000   0.00000E+00   0.00000E+00
*                           48   95.0000   0.00000E+00   0.00000E+00
*                           49   97.0000   0.00000E+00   0.00000E+00
*                           50   99.0000   0.00000E+00   0.00000E+00
*
*                          ...zone edit for the sum of the neutron groups ...
*
*
*                             ... constituent isotope   6 - pu-239   ...
*                       zone     volume      nusigf        fiss
*                          1 5.0265E+03   8.93230E-01   2.92476E-01
*                          2 1.0367E+04   0.00000E+00   0.00000E+00
*                          3 1.6022E+04   0.00000E+00   0.00000E+00
*                        ---  ---------   ----------   ----------
*                        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   0.00000E+00   0.00000E+00
*                        ---  ---------   ----------   ----------
*                        sum 3.1416E+04   1.00172E-01   3.33907E-02
*****************************************************************************************************************************
```

run highlights

*xxy start run highlights *

solver module execution suppressed.
interface file editit written.
start edit execution.
edits completed.

storage and timing history

module	scm words	scm limit	lcm words	lcm limit	cpu seconds	sysio seconds
0	0	0	0	0	1.3	0.0
100	3706	40000	0	0	1.0	0.0
101	0	0	0	0	0.6	0.0
102	0	0	0	0	0.0	0.0
103	0	0	0	0	0.0	0.0
104	0	0	0	0	0.0	0.0
105	0	0	0	0	0.0	0.0
106	0	0	0	0	0.0	0.0
107	0	0	0	0	0.0	0.0
108	0	0	0	0	0.0	0.0
109	3722	40000	0	0	0.1	0.0
300	0	0	0	0	0.4	0.0
301	1208	40000	289	140000	0.2	0.0
302	0	0	0	0	0.0	0.0
400	0	0	0	0	0.0	0.0

APPENDIX B: OPERATING SYSTEM SPECIFICS

UNIX/UNICOS Execution

On UNIX or UNICOS systems, the input is on STDIN and the printed output is on STDOUT. Thus, the user will normally cause execution of the program with the command:

dant.x < *odninp* > *odnout*

where - dant.x is the name of the executable file, *odninp* is the user's choice for a name for the input file, and *odnout* is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is dant.x.

STDERR contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDERR are any error messages.

Library Search Path

Most files read or written by ONEDANT are in the current UNIX working directory. Some forms of cross-section files may be kept in other directories. By setting the environment variable SNXSPATH, the user may specify an ordered set of alternate directories in which the program should look for the named files. As an example, if an ISOTXS file is in the directory, `/usr/tmp/xs`, then the following command can be used

```
setenv SNXSPATH /usr/tmp/xs
```

and ONEDANT will then look in that named directory for the library. The search path for each of the possible libraries is given in Table 2.2.

Table 2.2 UNIX Search Path

LIB	SEARCH PATH
MACRXS	Current Working Directory (CWD).
GRUPXS	**SNXSPATH**, then CWD.
ISOTXS	**SNXSPATH**, then CWD.
BXSLIB	**SNXSPATH**, then CWD, but see text below.
ODNINP	None, the library is contained in the input file.
MACBCD	CWD
XSLIBB	CWD
MENDF[a]	Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.
MENDFG[b]	
XSLIB	**SNXSPATH**, then CWD
other	For any name other than those above, the program will assume the form is XSLIB and search for it in **SNXSPATH**, then CWD.

a. Available only at Los Alamos.

b. Available only at Los Alamos.

SNXSPATH can be used to protect an input BXSLIB file from being overwritten. See the discussion on page 2-38.

TWODANT USER'S GUIDE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

3

XTM — Transport Methods Group

Los Alamos
National Laboratory

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

An Affirmative Action/Equal Opportunity Employer

This work was supported by the US Department of Energy.

USER'S GUIDE FOR TWODANT: A CODE PACKAGE FOR TWO-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT

by
Ray E. Alcouffe, Randal S. Baker, Forrest W. Brinkley,
Duane R. Marr, and R. Douglas O'Dell

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The TWODANT code is a modular computer program designed to solve the two-dimensional, time-independent, multigroup discrete-ordinates form of the Boltzmann transport equation.[1,2] It is a two-dimensional version of the one-dimensional code, ONEDANT.[3]

TWODANT™ is based on the modular construction of the DANTSYS™ code system package. This modular construction separates the input processing, the transport equation solving, and the postprocessing, 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 used by all the codes in the package. Different solution techniques are invoked simply by executing different SOLVER modules in the package. This SOLVER choice is automatically made by the INPUT module through an analysis of the input stream.

The TWODANT code is simply the DANTSYS code system package with a two-dimensional SOLVER module.

Some of the major features included in the TWODANT package are:

1. a free-field format text input capability, designed with the user in mind,

2. standardized data and file management techniques as defined[4] and developed by the Committee on Computer Code Coordination (CCCC); both sequential file and random-access file handling techniques are used,

3. the use of a diffusion synthetic acceleration scheme[5] to accelerate the iterative process in the SOLVER module,

4. direct (forward) or adjoint calculational capability,

5. x-y, r-z, and r-theta geometry options,

6. arbitrary anisotropic scattering order,

7. vacuum, reflective, periodic, white, or surface source boundary condition options,

8. inhomogeneous (fixed) source or k-effective calculation options, as well as time-absorption (alpha), nuclide concentration, or dimensional search options,

---

DANTSYS and TWODANT are trademarks of the Regents of the University of California, Los Alamos National Laboratory.

9. "diamond-differencing" for solution of the transport equation,

10. a diffusion solver that uses the multigrid method,[6]

11. user flexibility in using either text or sequential file input,

12. a Monte Carlo/discrete-ordinates (MC/S_n) option for solving all or part of the problem geometry using Monte Carlo,[7]

13. a ray trace first collision option to obtain a first collision source from an arbitrary source distribution (volume and external boundary sources),

14. user flexibility in controlling the execution of both modules and submodules, and

15. extensive, user-oriented error and warning diagnostics.

DOCUMENTATION FOR TWODANT USAGE

The documentation described here constitutes a complete manual for the use of the TWODANT code. It is intended to fully replace the former TWODANT manual.[8]

Included are two general categories of information. The first category is in this User's Guide and is oriented towards preparing input to the code. The second category is of a background, reference, conceptual, or theoretical nature and is intended primarily for the novice or first time user; an experienced user generally needs only this User's Guide.

What Is In This User's Guide

This User's Guide is a chapter from the much larger DANTSYS document. This Guide provides the ASCII text input specifications for TWODANT.

The guide is intended to serve as a complete input manual for two classes of user. Special, succinct sections containing summaries and compact tables are intended for the advanced user in order to make his input preparation more efficient. The main body of the guide concerns itself with descriptions of the input and should be sufficient for the user familiar with discrete ordinates concepts. Novice users may find other chapters of the document necessary.

This Guide first gives an overview of the input block order required by the code.

Next is a "mini-manual" in which are listed all the names of available input arrays arranged by input block. Definitions of input arrays are not given, as the names are suggestive, but expected types and sizes are provided. This mini-manual is very useful to the user as a quick check for completeness, a quick reference to type and size, and as an index into the more detailed array descriptions that follow. For the experienced user, the mini-manual is frequently all that is needed to prepare a complete input deck.

Following the mini-manual are reference sections describing in detail all the input parameters and arrays.

Appendix A provides two sample TWODANT cases with model, input, and output descriptions.

Lastly, Appendix B details operating system specifics, including how to effect an execution of the code.

Information of a reference, background, or theoretical nature that the first time user may need may not be found in this User's Guide, but the user will encounter liberal references to other chapters of this document for that sort of information.

What Is Available Elsewhere

In addition to this User's Guide, the user, especially the first time user, may find the information below described in other chapters of this document pertinent. For even greater detail on some of the general items, particularly the methods items, the user should look at Ref. 9.

The chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1 discusses in more detail the geometry and solver concepts and their related input. If the User's Guide proves insufficient for your needs, look in this chapter. Among the many sections of the chapter are ones on the input of inhomogeneous sources and a discussion of eigenvalue searches. There is also more detail on the Block-I input.

A discussion of how the EDIT module works and more detail on preparing the input is given in the chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 serves as the reference manual for the free-field input (rules, format, and operators) used in this code. That chapter is summarized in this guide, but should the summary prove inadequate, the user is referred there for full details.

The chapter "CROSS-SECTION LIBRARIES" starting on page 10-1 gives details of the many library formats available to TWODANT, including sections on how to prepare your own card-image (or text) libraries.

The chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1 describes the mixing concepts in detail and shows some examples.

Next is the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. That chapter describes the theoretical basis for the TWODANT code as well as the other codes in the DANTSYS package.

In the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure" starting on page 13-1 is shown a brief 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. In particular, the code package structure must be understood in order to make up input for piecewise executions of the code that are possible with controls that are part of the input in Block-I.

Error diagnostics that the user might encounter are found in the chapter "ERROR MESSAGES" starting on page 14-1. Several examples of input errors and the resulting error messages are provided for the user.

The chapter "FILE DESCRIPTIONS" starting on page 15-1 is a reference that describes all the files used by the package. Included is a detailed description of the file structure of the code dependent, binary, sequential interface files generated by and used in the

DANTSYS package. Also included are descriptions of any other files produced or used by the package, both binary and text. In some cases, this may simply be a reference to a more comprehensive document, such as the file descriptions for the CCCC standard interface files.

TWODANT INPUT OVERVIEW

Input Block Order

The full TWODANT input consists of a title line section, followed by six blocks of free field input. The title line section is not free field. Any input referred to as a block uses the free field input form.

Block-I consists of basic control and dimensional information that allows efficient packing of the array data. This information also allows checking of the lengths of arrays supplied by interface files.

Block-II contains the geometric information.

Block-III consists of the nuclear data specifications.

Block-IV contains mixing information.

Block-V contains the rest of the input needed for specifying the flux calculation.

And lastly, Block-VI contains the edit (i.e., report writing) specifications.

If a text cross-section library is to be included in the input deck, it should be placed between Blocks III and IV. TWODANT supports many library formats, and so the library may or may not be in free field format depending upon the option chosen.

A full input would then look like that diagrammed on the following page.

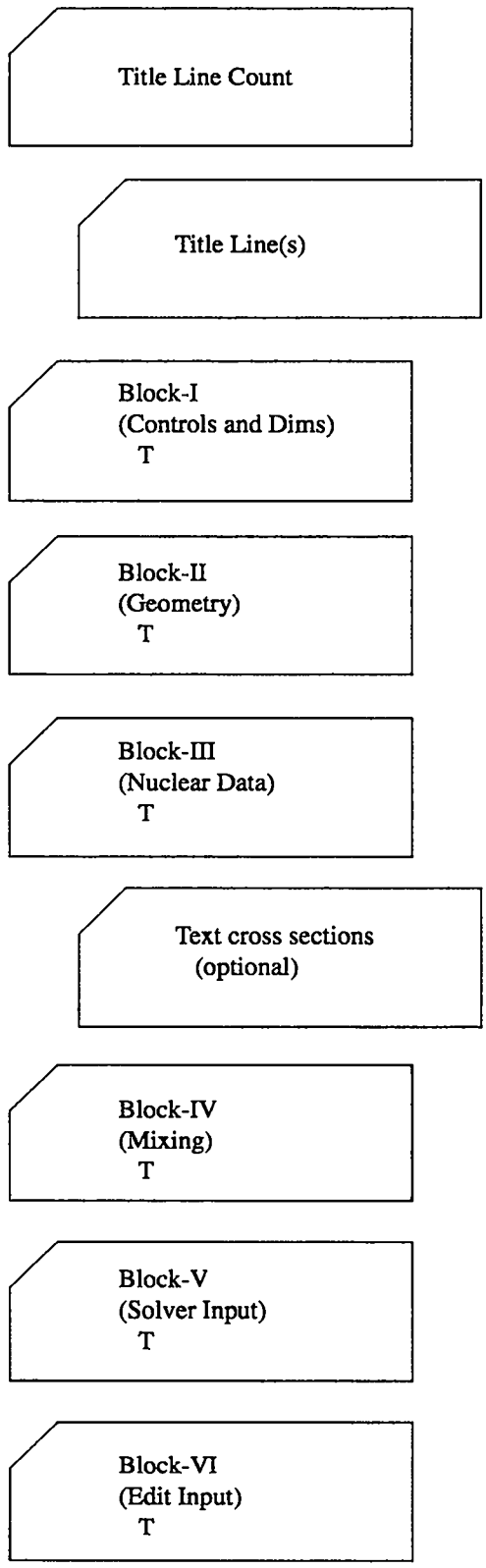


Figure 3.1 TWODANT Input Order

Free Field Input Summary

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 is summarized here for quick reference.

There are four basic input quantities in the free field input used in TWODANT; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is briefly described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array." To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, might look like:

```
SOURCX= 0 0 0 1.1 1.1 0 0 0 0 0
```

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Single value input variables are treated as arrays of unit length.

Numeric Data Items

Numeric data items follow a Fortran input convention. For example, all of the following are valid entries for the number ten:

```
10, 1.0+1, 1E1, 10.0
```

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a Fortran variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters and blanks may be included if the data item is surrounded by double quotes. Operators may NOT be used with character data items.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this, the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described later, in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read, no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input.

The data operators are specified in the general form

n *O* d

where:

n is the "data numerator," either an integer or a blank;
O is any one of the "data operator" characters shown below; and
d is a "data entry" (may be blank for some operators).

Note: The "data operator" character must be appended to the "data numerator."

Using operators, the SOURCX input described above could more succinctly be given as:

SOURCX= 0 0 0 2R 1.1 F0

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO[10] operators may be used in numeric entry.

A table of the most used operators is given next including brief descriptions. For full descriptions of these and a complete list of all the available operators, including the more esoteric ones, the user is referred to "FREE FIELD INPUT DETAILS" on page 9-13.

Frequently Used Operators

Operator[a]	Functionality
nR d	REPEAT the data item d, n times.
nI d	INTERPOLATE (linear) n data items between data item d and the next data item.
nC d	SCALE (multiply) the n previous entries by d.
F d	FILL the rest of the data string with the data item d.
nY m	STRING REPEAT. Repeat the previous m strings, n times.
nL d	INTERPOLATE LOGARITHMICALLY n data items between d and the next d.
nZ	ZERO. Enter the value zero n successive times.
nS	SKIP. Skip the next n data items.
nQ m	SEQUENCE REPEAT. Enter the last m entries, n more times.
nG m	SEQUENCE REPEAT WITH SIGN CHANGE. Same as the Q option but the sign of the m entries is changed every repeat.
nN m	SEQUENCE REPEAT INVERT. Same as the Q option but the order of the m entries is inverted each repeat.
nM m	SEQUENCE REPEAT INVERT WITH SIGN CHANGE. Same as N option but the sign is also changed every repeat.
nX	COUNT CHECK. Causes code to check the number of entries in the current string so far, against the number n.

a. The operator character must always be appended directly to n. d or m need not be immediately adjacent to the operator character.

MINI-MANUAL Introduction

On the following few pages is given a complete list of the input names, expected array sizes, and order within the array. No description of the array contents is given in this MINI-MANUAL as full details are given in later sections. The MINI-MANUAL is intended to serve as a quick reference for the knowledgeable user.

In both the MINI-MANUAL and in the detailed sections which follow, a shorthand form is used to indicate the size and order of the array that the code expects. This information is enclosed in square brackets immediately after the array name. Essential features are:

1. A single entry in the brackets is the array length.

2. No brackets at all indicates a simple variable (i.e., an array of unit length).

3. A dash (-) in the brackets indicates an arbitrary length.

4. A semicolon (;) indicates that the input for that array is expected in strings. To the left of the semicolon is the string length. To the right of the semicolon is the number of strings in the array.

5. If the number of strings is shown as a product, the order is important. The leftmost quantity must be exhausted first, then, the next one to the right is varied. For example, the array name for the full spatial source distribution is shown as:

   SOURCF [IT;JT*NMQ]

   where - IT is the number of fine meshes in the X-direction, JT is the number of fine meshes in the Y-direction, and NMQ is the number of input source moments. For this array, the first string is composed of the P_0 source values for each x mesh point in the first y mesh. The next string is the P_0 source values in the second y mesh. This process is repeated for all JT y meshes. Then starting again with the first y mesh, the P_1 source values for each x mesh are given. After all P_1 values are given, the P_2 values follow. Continue until all NMQ moments are specified.

Note: Usually, values for the quantities within brackets will have already been specified in the input. Sometimes, however, a quantity is derived from the array input itself. For instance, in this particular case, NMQ is not an input quantity; rather, the code counts the number of strings and then, knowing JT, deduces what NMQ must have been.

MINI MANUAL

Title Line Control
(3I6 Format)
NHEAD,NOTTY,NOLIST

Title Line(s)

(IF NHEAD>0)

Block-I:Controls & Dimensions

IGEOM
NGROUP
ISN
NISO
MT
NZONE
IM
IT
JM
JT

MAXLCM
MAXSCM

-- -- -- -- --

MINIPRT

NOSOLV
NOEDIT

NOGEOD
NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM

T

Block-II:Geometry

XMESH [IM+1]
YMESH [JM+1]

XINTS [IM]
YINTS [JM]

ZONES [IM;JM]

T

Block-III: Cross Sections

LIB
  valid: *ODNINP*
  XSLIB
  ISOTXS
  GRUPXS
  BXSLIB
  MACRXS
  MACBCD
  XSLIBB
  (local)*MENDF*
  (local)*MENDFG*
  alternate XSLIB name

WRITMXS
  valid: *MACBCD*
  XSLIBB
  XSLIBF
  XSLIBE

LNG
BALXS
NTICHI
CHIVEC [NGROUP]
LIBNAME
-- -- -- -- -- -- --
Rest of this block is needed only for text libraries.

MAXORD
IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD (default:1)
NAMES [NISO]
EDNAME [IHT-3]
NTPI [NISO]
VEL [NGROUP]
EBOUND [NGROUP+1]

T

iff LIB= ODNINP, insert ASCII text cross sections here

Block-IV: Mixing

MATLS [-;MT]
ASSIGN [-;NZONE]

PREMIX [-;-]

ASGMOD [-;-]
CMOD

FMMIX

MATNAM [MT]
ZONNAM [NZONE]
MATSPEC [-]
  valid: *ATFRAC*
  WTFRAC
  ATDEN

ATWT [-]

T

Block-V: Solver

IEVT
ISCT
ITH
IBL
IBR
IBT
IBB

EPSI
IITL
IITM
OITM
ITLIM
NOSIGF

KCALC

Solver (continued)

--- Output Controls ---

FLUXP
XSECTP
FISSRP
SOURCP
ANGP
BALP
RAFLUX
RMFLUX

ASRITE
ASBOTT
ASTOP
ASLEFT

--- Miscellaneous ---

TRCOR
valid: *DIAG*
BHS
CESARO
NO
NORM
BHGT

CHI [NGROUP;M]

DEN [IT;JT]
-or-
DENX [IT], DENY[JT]

WDAMP [NGROUP]

Solver (continued)

--- Quadrature -------

GRPSN [NGROUP]
IQUAD
WGT [MM]
MU [MM]
ETA [MM]

--- Flux Guess -------

INFLUX

--- Searches ---------

IPVT
PV
EV
EVM
XLAL
XLAH
XLAX
POD

XM [IM], YM [JM]

Solver (continued)

----Volumetric Source----

INSORS

SOURCE [NGROUP;NMQ]
-or-
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ]
-or-
SOURCE [NGROUP;NMQ] and
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ]
-or-
SOURCF [IT;JT*NGROUP*NMQ]
-or-
SOURCE [NGROUP;NMQ] and
SOURCF [IT;JT*NMQ]

-----Boundary Source-----

SILEFT [NGROUP;JT]
SIRITE [NGROUP;JT]
SIBOTT [NGROUP;IT]
SITOP [NGROUP;IT]
-or-

SALEFT [MM*2;NGROUP*JT]
SARITE [MM*2;NGROUP*JT]
SABOTT [MM*2;NGROUP*IT]
SATOP [MM*2;NGROUP*IT]
-or-

BSLFTG [NGROUP]
BSLFTY [JT]
BSLFTA [MM*2]

BSRITG [NGROUP]
BSRITY [JT]
BSRITA [MM*2]

BSBOTG [NGROUP]
BSBOTX [IT]
BSBOTA [MM*2]

BSTOPG [NGROUP]
BSTOPX [IT]
BSTOPA [MM*2]

Solver (continued)

---- First Collision Source ----

FCSRC
valid: *ANA*
RAY
NO
FCNRAY
FCNTR
FCWCO

---- Monte Carlo Option ----

MCOPT
MCREG [NGROUP]
MCIBND [2]
MCJBND [2]
MCBLT
MCITS
MCNHIS
MCNTR
MCISEED
MCIPRNT

---- Monte Carlo Biasing ----

MCSB
MCWC [2]
MCCMIMP [IM;JM]
MCEGIMP [NGROUP]
MCLVIMP [ISN/2]

---- Monte Carlo Sources ----

MCPTSRC [1/2]
MCBSLA [2]
MCBSRA [2]
MCBSBA [2]
MCBSTA [2]

T

Block-VI: EDIT

PTED
ZNED

POINTS [K], K≤IT*JT
EDZONE [IT;JT]

EDXS [K], K≤NEDT
RESDNT
EDISOS [K], K≤NISO
EDCONS [K], K≤NISO
EDMATS [K], K≤MT
XDF [IT]
YDF [JT]

RSFE [NGROUP;-]
RSFX [IT;-]
RSFY [JT;-]
RSFNAM [-]

ICOLL [K], K≤NGROUP
IGRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

PRPLTED

T

TWODANT INPUT DETAILS

Introduction

The following pages of this section give details for each of the input arrays. All valid TWODANT arrays are discussed in this section in detail complete enough to form the input.

However, the beginning user, particularly one unfamiliar with discrete-ordinates codes, may find that he is missing some information of a background nature. See "What Is Available Elsewhere" on page 3-14 for that.

First, here are a few general instructions:

1. All six of the input blocks are normally included. Block-I is always required but any of the other five blocks may be omitted under the proper conditions. The input module reads each block in turn and from it generates one or more binary interface files. The interface files drive the SOLVER and EDIT modules. Thus, if the user wants no edits, the Block-VI input may be omitted. Then with no interface file, the EDIT module will not be executed. Alternatively, if the interface file is available from another source, the corresponding block of input may be omitted. For instance, Block-II describes the geometry. The input module normally writes this information to the GEODST interface file. If the GEODST file is available from another source or a previous run, the Block-II input may be omitted.

2. A general theme of the TWODANT input is that arrays that are not needed are not entered. Presence of an array indicates that it should be used. Thus, for example, if the density array is entered (DEN array), the cross section at each mesh interval will be modified accordingly. No separate switch need be set to say that the calculation should be done. To eliminate the density modification, simply remove the DEN array from the input or comment it out.

3. The arrays, in general, are grouped in the input instructions according to function. Thus, for example, the input arrays for the volumetric source are found in a single table, or grouping, of input.

4. Groupings of input data may be marked as "Required" or "Optional" in order to guide the user and speed navigation through the input instructions.

   "Required" means that at least one of the arrays in the grouping must be entered. Thus, you must read through the grouping and enter at least one of the arrays found there.

Groupings marked "Optional" may be skipped if the subject is inappropriate. Thus, using the previous example, if one has no volumetric source, one simply skips to the next grouping of input; there is no need to read about any of the arrays within the volumetric source grouping.

Arrays in groupings not marked as "Required" or "Optional" should be reviewed. These groupings contain arrays of vital data that are used in every calculation, but have default values. Thus, although you may not make any input to these arrays and they are in that sense optional, you must concern yourself with them to ensure that the default values are what is intended.

5. Input arrays may also be marked individually. If not marked, they inherit the marking of the grouping in which they are contained. Thus, an unmarked array in a "Required" grouping is required input and you must enter that array. An unmarked array in an "Optional" grouping is optional.

   You may encounter a "Required" array within an "Optional" grouping. That means that if you decide to invoke the option represented by that grouping, you must input that particular array. For example, if you want user defined response function reaction rates calculated, you must input the RSFE array.

   All arrays within unmarked groupings are optional. However, values in these arrays may be used by the code, so you should concern yourself with the default values if you choose not to enter a value.

6. Unless specifically noted otherwise, the default on all numeric inputs is zero.

7. In an adjoint run, none of the groupwise input arrays should be inverted. The code will externally identify all groups by the physical group number, not by the calculational group number (the calculational group number is in inverse order). Thus, the user interface should be consistently in the physical group order.

8. The use of information within square brackets to indicate the size of arrays and strings and the order within those arrays is the same as described in "MINI-MANUAL Introduction" on page 3-22.

9. Except where noted, arrays and strings must contain the exact number expected by the code (as indicated in the array or string description). If not, the code will eventually abort with a (hopefully) descriptive error message or messages.

10. New users reading these instructions for the first time and unfamiliar with the TWODANT input may find it helpful to follow the sample input in Appendix A while reading this section.

11. Array names are shown here in upper case. What you should actually input for them will depend upon the code's implementation on your platform. At the present time, on most platforms, you should use lower case input.

12. Items in italics in the input instructions indicate actual values that may be entered for an array. You will frequently find switches where the input is the digit 0 or the digit 1. This will be represented by *0/1* in the input description. In other arrays where an exact character string is required such as "ISOTXS" in the LIB array, you will find the notation *ISOTXS*. Note that in this notation, the word is both upper case and italicized. This combination means you must enter exactly those characters. Again, although the characters will be shown here in upper case, what you should actually input for them will depend upon the code's implementation on your platform.

13. When a template for the input form is given, as for the MATLS array, the style in the template tells the user what is expected. If an input word or value is lower case and italicized, the user is to replace that position with the entry of his choice. If the input word is in italicized style and in upper case, the user is to input exactly those characters to achieve the desired result. Depending on the implementation on your platform, the input word, itself, is usually in lower case.

14. Units to be used for the input quantities are not spelled out as they only need to be self consistent. However, the following are commonly used: Dimensions in centimeters, isotopic cross sections in barns per atom; then it follows that atom densities are in atoms per barn-centimeter. Sources are particles per cm^3 per second for volumetric sources and particles per cm^2 per second for boundary sources; fluxes will then be in particles per cm^2 per second.

Title Line Details

Title Line Control (format 3I6)[a] {Required}

Word	Name	Comments
1	NHEAD	Number of title lines that follow.[b]
2	NOTTY	Suppress output to on-line user terminal? *0/1* = no/yes.
3	NOLIST	Suppress listing of all ASCII text input? *0/1* = no/yes. (default=no)

a. WARNING! Note that this first line is in fixed format.

b. Follow this control line with NHEAD title lines containing descriptive comments. Each title line may contain up to 72 characters.

Block-I Details: Dimensions and Controls

Dimensions {Required}

Name	Comments
IGEOM	Geometry. *6/7/11* = x-y/r-z/r-theta or use one of the following character strings: *X-Y, R-Z, R-THETA*
NGROUP	Number of energy groups.
ISN	S_n order to be used. If ISN is negative, code will use the Chebychev-Legendre (IQUAD=2) quadrature set. (See IQUAD input in "Quadrature Details" on page 3-54).
NISO	Number of isotopes on the basic input cross-section library.
MT	Number of materials[a] in the problem (mixed from the isotopes).
NZONE	Number of geometric zones[b] in problem.
IM	Number of coarse mesh intervals[c] in the x (or r) direction.
IT	Total number of fine mesh intervals[d] in the x (or r) direction.
JM	Number of coarse mesh intervals in the y (or z) (or theta) direction.
JT	Total number of fine mesh intervals in the y (or z) (or theta) direction.

a. Material is defined on page 3-44.

b. Zone is defined on page 7-13.

c. Coarse mesh is defined on page 7-13.

d. Fine mesh is defined on page 7-13.

Storage Requirements {Optional}

Name	Comments
MAXSCM	Length of SCM desired (default=40000_{10})
MAXLCM	Length of LCM desired (default=140000_{10})

The above input (Dimensions plus Storage Requirements) for Block-I will cause the code to attempt to produce a full run, subject to availability of the input normally found in the other Blocks. The controls below allow shortened print files, partial runs (say, of only the input module), or cause the code to ignore any of the other input Blocks present. For full details on their use, see "PIECEWISE EXECUTION" on page 13-19.

Minimum Print Output {Optional}

Name	Comments
MINIPRT	Provide a shortened output print file.[a] *0/1 = no/yes*[b]

a. This shortened print file contains the title, the name of the cross-section library used, the core storage needed, the iteration monitor, the global balance table, and not much else. However, existing print controls, such as the variable NOLIST on the first line which controls the listing of the input lines and the array print controls in the solver modules, work as before and are not affected by the MINIPRT input. The EDIT module also works as before; MINIPRT has no effect on its output.

b. You may also use the words no or yes as an entry.

Run Configuration Controls {Optional}

Name	Comments
NOSOLV	Suppress solver module execution. *0/1* = no/yes.
NOEDIT	Suppress edit module execution. *0/1* = no/yes.
NOGEOD	Suppress writing GEODST file even though the geometry input (Block-II) may be present. *0/1* = no/yes.
NOMIX	Suppress writing mixing files even though the mixing input in Block-IV may be present. *0/1* = no/yes.
NOASG	Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. *0/1* = no/yes.
NOMACR	Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. *0/1* = no/yes.
NOSLNP	Suppress writing the SOLINP file even though Block-V may be present. *0/1* = no/yes.
NOEDTT	Suppress writing the EDITIT file even though Block-VI may be present. *0/1* = no/yes.
NOADJM	Suppress writing the ADJMAC file even though an adjoint calculation is called for. *0/1* = no/yes.

Note: Default on all these controls is no.

Block-II Details: Geometry

Geometry Arrays[a] {Required}

Name	Comments
XMESH [IM+1]	x (or r) coordinates of coarse mesh edges.
YMESH [JM+1]	y (or z) (or theta[b]) coordinates of coarse mesh edges.
XINTS [IM]	Number of fine meshes in each coarse x (or r) mesh.
YINTS [JM]	Number of fine meshes in each coarse y (or z) (or theta) mesh.
ZONES [IM;JM]	Zone number[c] for each coarse mesh. This array defines the geometric zones to which cross-section materials are assigned. The zone number must not be greater than NZONE.

a. Definitions of coarse mesh, fine mesh, and zone are given in the chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1. See note on units on page 3-31. The information entered in this block is written to the CCCC standard interface file GEODST.

b. The units of theta are revolutions.

c. A zone number of zero indicates the mesh contains a void, and no cross section will be associated with that mesh. The zero zone number is not counted in the total zone count NZONE.

Block-III Details: Nuclear Data

Nuclear Data Type and Options {Required}

Name	Comments	
LIB	Name[a] and form of the cross-section data file. Enter as a data item one of the following words:	
	Word	Description
	ISOTXS[b]	CCCC standard isotope ordered binary cross-section file.
	XSLIB	ASCII text library supplied in a separate file named XSLIB.
	ODNINP	ASCII text library follows after this block of input (after the T of Block-III).
	GRUPXS[c]	CCCC standard group ordered cross-section file.
	BXSLIB	Binary library supplied as a separate file named BXSLIB. [See "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 10-12.
	MACRXS[d]	Use existing files named MACRXS for SOLVER module, SNXEDT for EDIT module. These files were created in a previous run. 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.
	XSLIBB	See "XSLIBB Card-Image Library File" on page 10-12.
	MACBCD	ASCII form of MACRXS file.
	MENDF	(LANL only) See "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.
	MENDFG	(LANL only) See "The Los Alamos MENDF5G Gamma Cross-Section Library" on page 10-14.
	other	If a word other than those listed above is entered, the code will use the file with that word as its name, provided that file exists in the user's file space. Such a file must be structured as an XSLIB file.
WRITMXS {optional}	Controls the code's writing certain ASCII cross-section files.[e] Enter one of the following words:	

Nuclear Data Type and Options (Cont.) {Required}

Name	Comments
	Word — Description
	MACBCD — Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file.
	XSLIBB — Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file.
	XSLIBE — Creates the cross-section file named XSLIBE, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBE is in Los Alamos 6E12 format (IFIDO=0).
	XSLIBF — Creates the cross-section file named XSLIBF, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBF is in FIDO fixed-field format (IFIDO=1).
LNG {optional}	Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.
BALXS {optional}	Cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using! Value — Description *-1* — balance cross sections by adjusting absorption cross section. *0* — do not balance cross sections. (default) *1* — balance cross sections by adjusting self-scattering cross section.
NTICHI {optional}	MENDF fission fraction to be used for the problem (LANL only). *1/2/3* = Pu239/U235/U238 (default is U235). Will be overridden by any CHIVEC input described below or by any zone-dependent CHI in input Block-V.
CHIVEC [NGROUP] {optional}	Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.

a. On UNIX systems, the user may specify a search path for some of these files using the environment variable SNXSPATH. See "Library Search Path" on page 3-124 for details.

b. The CCCC standard for file ISOTXS does not allow the inclusion of the 2L+1 term in the higher order scattering cross section. However, if you have a nonstandard file which contains the 2L+1 term, you may override by setting I2LP1=1. See "Text Cross-Section Library Format" on page 3-41. TWODANT will then convert the cross sections to the appropriate internal form.

c. The 2L+1 term on GRUPXS is treated the same as for ISOTXS. See footnote b.

d. In the convention used in this user's guide, a MACRXS library contains "material" cross sections; all the other libraries contain "isotope" cross sections.

e. See "COUPLED NEUTRON-GAMMA CROSS SECTIONS" on page 10-15.

Alternate Library Name {Optional}

Name	Comments
LIBNAME	Alternate name of the library file. May be used only with certain types of libraries. See Table 3.1.

The entries in the LIB input variable normally dictate both the form and the name of the cross-section library. If the user specified ISOTXS, for example, the code would look for a file named ISOTXS and expect it to be in the CCCC format for an ISOTXS file.

For some libraries, the user may specify the form in the LIB array and specify separately the name in the LIBNAME array. The libraries that can be treated this way are shown in Table 3.1.

Table 3.1 LIBNAME Availability

LIB	LIBNAME AVAILABLE?
MACRXS	No
GRUPXS	Yes
ISOTXS	Yes
BXSLIB	Yes
ODNINP	No
MACBCD	No
XSLIBB	No
MENDF[a]	No
MENDFG[b]	No
XSLIB	Yes
other	Ignored

a. Available only at Los Alamos.

b. Available only at Los Alamos.

The BXSLIB file requires special treatment. It is normally created when the original library is a text library in the ODNINP or XSLIB form. In subsequent runs, this binary BXSLIB file may be used as the source of the cross-section data. The user may wish to save this file under another name. The program, in future runs, may then access the library for reading by using LIBNAME to specify that name.

This procedure is wise because some cases using the BXSLIB form as input also require rewriting it in order to add new information. When this situation arises, the rewritten file is always named BXSLIB. Thus, if the original BXSLIB form library had a different name, it would be protected from being overwritten. For the remainder of the current run, the program will access the file named BXSLIB.

Text Cross-Section Library Format {Required if LIB= XSLIB or LIB=ODNINP}

Name	Comments
MAXORD	Highest Legendre order in the scattering tables.
IHM	Number of positions (entries) in each row of the cross-section table.
IHT	Position number of the total cross section.
IHS {optional}	Position number of the self-scatter cross section. (default = IHT + 1).
IFIDO {optional}	Format of the cross-section library. *-1/0/1/2* = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.
ITITL {optional}	A title line precedes each table. *0/1* = no/yes
I2LP1 {optional}	Higher order scattering cross sections on the library contain the 2L+1 term. *0/1* = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.
SAVBXS {optional}	Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. *0/1* = no/yes.
KWIKRD {optional}	Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? *0/1* = no/yes (default=yes).
NAMES [NISO] {optional}	Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).
EDNAME [IHT-3] {optional}	Character name for each of the EDIT cross-section positions used in the cross-section edits. These are the positions before the absorption cross section in the cross-section table. (default names are: EDIT1, EDIT2, . . . etc.).
NTPI [NISO] {optional}	Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).
VEL [NGROUP] {optional}	Speeds for each group. Needed only for alpha calculations.
EBOUND [NGROUP+1] {optional}	Energy boundaries for each group.

ASCII text libraries may be entered in one of the four forms indicated by the IFIDO input. All four forms share the following features: Cross sections are entered in a table optionally preceded by a title line. A table consists of NGROUP rows of entries. Each row contains the cross sections for a single group and consists of IHM entries. The user specifies the positions in the row occupied by the total and selfscattering cross sections. Order within a row (e.g., for group g) is then as follows:

$$\ldots \sigma_{abs},\ \nu\sigma_{f},\ \sigma_{total},\ \ldots \sigma_{g+2\rightarrow g},\ \sigma_{g+1\rightarrow g},\ \sigma_{g\rightarrow g},\ \sigma_{g-1\rightarrow g},\ \sigma_{g-2\rightarrow g},\ \text{etc.}$$

Notice that all terms in the scattering matrix are in positions relative to that of the self-scattering position and the rest of the cross sections are in positions relative to the position of the total cross section. The positions before the absorption cross section are frequently used for edit cross sections. For more detail, see "Ordering of Cross Sections Within a Cross-Section Table" on page 10-10.

Different Legendre orders are in different tables, which follow in order.

The user may order the group structure either by increasing energy or by decreasing energy. However, it is conventional and desirable for most problems to order it by decreasing energy, that is, group one is the highest energy. In that case, the scattering cross sections to the left of $\sigma_{g\rightarrow g}$ such as $\sigma_{g+1\rightarrow g}$ are upscattering terms and the terms to the right of $\sigma_{g\rightarrow g}$ are the downscattering terms.

In the Los Alamos format, the table is entered with a standard Fortran 6E12 format.

For greater precision in your input, use the 4E18 option.

In the fixed field FIDO format that ANISN[10] uses, entries are made in six twelve-column fields. Each twelve-column field is divided into three subfields, a two-column numeric field, a one-column character field, and a nine-column numeric field. See page 9-19 for details if you are not familiar with this input. The last field in each table must have the character T in the character position. No array identifier should be used. This format also restricts the usable input operators to T, *, R, -, +, and Z.

In the free field form, entries do not have to be in designated columns. Rather, the rules specified in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 apply. Each table in this form is also terminated with the character T. No array identifier (i.e., array name with appended equals sign) should be used.

Block-IV Details: Cross-Section Mixing

A short summary of the primary mixing arrays, MATLS and ASSIGN, is given here for quick reference. Normally, THESE TWO ARRAYS ARE REQUIRED and, in most problems, would be the only arrays in this block. Other mixing arrays are also briefly described.

There are actually several nested levels of mixing. Each level has the job of calculating values from expressions of the form: $\Sigma_g = \sum_{i=1}^{k} N_i \sigma_{i,g}$ for each group, g. The user's job is to input the N_i for all the k components of the mixture and to specify each component, i. Component i has the cross section, $\sigma_{i,g}$. In common usage, for the first level of mixing, $\sigma_{i,g}$ is the effective microscopic cross section and N_i is the atom density of isotope i, and Σ_g is then the macroscopic cross section of some material. In a higher level of mixing, these materials may be homogenized into a single material by using their volume fractions for the N_i. With several nested levels, the user has a great deal of flexibility in defining what Σ_g is for that level. A more complete discussion of mixing will be found in the chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1.

A discussion of cross section processing is outside the scope of this document, but it should be noted that the user needs to be aware of the processing that is inherent in the input library. For instance, for materials in which there are isotopes with cross-section resonances, self shielding of the cross sections for these isotopes may be important and this effect must have been considered in the preparation of the "effective" microscopic cross sections for these isotopes. Since the self shielding is dependent on the amounts and types of the other isotopes in the material, the "effective" cross section is strictly valid only for use in a mixture which has the same composition as was used in the self shielding calculation. If the user desires to use this same "effective" microscopic cross section in some other composition (mix) of material, it is up to the user to verify the accuracy of this approach.

Primary Mixing Arrays {Required}

Name	Description
MATLS[a] [-;MT]	Instructions for mixing "isotopes" or premixes into "materials." See details below.
ASSIGN[b] [-;NZONE]	Assignments of materials to geometric zones. See below.
PREMIX [-;-] {optional}	Instructions for mixing "isotopes" into premixes. See below.

a. The information entered in the MATLS array is written to the CCCC standard interface files NDXSRF and ZNATDN.

b. Information entered in the ASSIGN array is written to the code-dependent interface file ASGMAT.

In order to understand how cross sections are mixed and the resultant material placed in the problem, we first need a little conceptual information.

The key entities used in specifying the cross-section spatial distribution are coarse mesh, zone, isotope, and material.

The basic geometry of the problem is defined with the coarse meshes specified in Block-II. The geometric areas called zones are also defined there using the ZONES array; the ZONES array designates the zone number assigned to each coarse mesh.

Here in Block-IV, we mix cross sections and assign them to the zones created in Block-II. For the purposes of this discussion, the cross sections found on the input library belong, by definition, to "isotopes", no matter what their true nature. These "isotopes" may then be mixed to form materials, using the MATLS array. Materials are then assigned to zones using the ASSIGN array.

MATLS input array

The general form of a MATLS mix instruction is shown below:

$$\text{MATLS=}\ \ mat_1\ \ comp_1\ den_1,\ \ comp_2\ den_2,\ \text{...etc....}\ ;$$

where mat_1 is the desired character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional materials (i.e., mat_2, mat_3, and so on up to the required number, MT) are defined in subsequent strings. Each string may contain as many components as necessary (actual limit = 500). A component is usually an isotope from the library, but may also be a temporary material created by the PREMIX array (see below).

When the component is an isotope, the den_i is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 3-49).

Short form: MATLS= *ISOS*

This form specifies that there should be as many materials as isotopes and that isotope number 1 is to be used for material number 1, isotope number 2 is to be used for material number 2, and so on.

In the special case where there is only a single component in a material and its density is unity, the density entry may be omitted as in the first material below:

MATLS= mat_1 $comp_1$; mat_2 $comp_2$ den_2; ...etc.... ;

ASSIGN input array

The general form of the ASSIGN instruction is shown below:

ASSIGN= $zone_1$ mat_1 vol_1, mat_2 vol_2, ...etc.... ;

where $zone_1$ is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). mat_1, mat_2, and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" vol_1, vol_2, and so on. Additional zones (i.e., $zone_2$, $zone_3$, and so on up to the required number, NZONE) are defined in subsequent strings. Although it is highly recommended that you use character names, here it is convenient to use the numeral for the zone name because it is the same numeral entered in the ZONES array.

Short form: ASSIGN= *MATLS*

This form specifies that there are as many zones as there are materials, and that material number 1 is to be assigned to zone number 1, material number 2 to zone number 2, and so on.

NOTE: The short form ASSIGN=*MATLS* can not be used if you intend to use the ASGMOD input array described later in this section.

PREMIX input array

The PREMIX array forms temporary materials in a way exactly analogous to the way that permanent materials are formed in the MATLS array. The difference in treatment is that the temporary materials created by PREMIX exist only long enough to complete the mixing; they are not available for assignment to geometric zones, nor are they available for use in material edits.

The general form of a PREMIX mix instruction is shown below:

PREMIX= $tmat_1$ $comp_1$ den_1, $comp_2$ den_2, ...etc.... ;

where $tmat_1$ is the character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional temporary materials (i.e., $tmat_2$, $tmat_3$, and so on) may be defined in subsequent strings. A component may be either an isotope from the library or another temporary material created by PREMIX.

The PREMIX array is useful for organizing the mixing input. For instance, it is frequently useful to mix the cross sections for a molecule of water and then in subsequent mix instructions, to input the molecular density of water as opposed to entering the atom density for both hydrogen and oxygen. Other examples are to form average cross sections for an element composed of many isotopes, or to form full density materials and then in later mix instructions to put in the volume fraction of the full density material.

Character Names vs. Numeric Names

In the foregoing discussion, isotopes, materials, and zones were identified by their character names. Optionally, they may be referred to by their ordinal number. Thus, 2 for an isotope name would call for the second isotope on the library. However, this practice is NOT recommended.

THE CHARACTER NAME FORM IS HIGHLY RECOMMENDED. It provides the most straightforward, most readable form. If the character name form is used, the naming input arrays in the following table are not needed.

Using the character name form in one array and the numeric name form in another array is particularly discouraged. However, should one wish to use the numeric form in the MATLS and/or ASSIGN arrays, and then subsequently associate character names with the ordinal numbers, one can use the naming arrays in the following table to do so. This situation could arise if, for some reason, one wanted to use material numbers in the MATLS array, but use character material names in the ASSIGN array.

When the library is of the MENDF form, the character names that must be used for the isotope names are discussed in "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.

Mixing Array for a Concentration Search {Optional}

Name	Description
ASGMOD[a] [-;-]	C_1 parameters used in concentration searches. See the discussion below.

a. The information entered in the ASGMOD array is written to the ASGMAT file together with the information from the ASSIGN and CMOD arrays.

ASGMOD input array

The ASGMOD array is used in conjunction with the ASSIGN array when one wishes to vary the composition of a zone or zones in order to achieve a certain value of k-effective or alpha (i.e., in a concentration search). The concentration (or volume fraction) of material x in zone z is given by the following expression:

$$C(z,x) = C_0(z,x) + C_1(z,x)*CMOD$$

where $C_0(z,x)$ is the base concentration of material x in zone z. This is the concentration (or volume fraction) entered in the ASSIGN array for material x. In these arrays, x is not any kind of an index; correspondence is made by name, rather than by position within the array. Thus, for instance, in a problem that had ten materials, we might only assign one of them to a given zone. It would then probably be in the first position in the ASSIGN array string for that zone even though it might have been say, sixth in the list of all materials.

$C_1(z,x)$ is the corresponding entry in the ASGMOD array for material x in zone z.

CMOD is the search parameter (sometimes called search eigenvalue) that will be varied by TWODANT in order to achieve the desired k-effective or alpha value. In a search calculation, the initial value for CMOD will be the input value EV.

The general form of the ASGMOD instruction is shown below:

ASGMOD= *zone* mat_m vol_m, mat_n vol_n, ...etc.... ;

where *zone* is the character name of any zone in the problem, mat_m, mat_n, and so on are the character names of any of the materials that will be present in this zone, and vol_m, vol_n, and so on are the C_1 values for respectively, mat_m, mat_n, and so on. Additional zones may be specified in subsequent strings. All zones do not have to appear in the ASGMOD array nor do all materials within a zone have to appear in the string for that zone.

Concentration Modifier {Optional}

Name	Description
CMOD	Concentration modifier. Input value is not used in a search. See the discussion below.

The concentration modifier, CMOD, is varied by TWODANT during a search calculation. For any other type of calculation, a value of CMOD may be input and the composition of the zones will be calculated using the expression above for C(z,x).

Fine Mesh Mixing[a] {Optional}

Name	Description
FMMIX	Read the composition of each fine mesh from the file LNK3DNT. *0/1* = no/yes.

a. x-y geometry only.

The fine mesh mixing algorithm is designed for a general geometry option in x-y geometry using a volume fraction method on the fine mesh. It implies that one has an auxiliary code which will generate the volume fraction data from a general geometry description and store it on the file LNK3DNT.

Miscellaneous Mixing Input {Optional}

Name	Comments
MATNAM [MT]	Character material names for Materials. Used only if the mat_1 name used in the MATLS array was integer. First entry in MATNAM array is the desired character name for Material number 1, second entry is the desired character name for Material number 2, etc.
ZONNAM [NZONE]	Character zone names for Zones. Used only if the zone name entry in the ASSIGN or ASGMOD array was integer. First entry in the ZONNAM array is the desired character name for Zone number 1, second entry is the desired character name for Zone number 2, etc.
MATSPEC [≤MT]	Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions. Allowable entries are the words: *ATDENS* (default) atomic densities *ATFRAC*[a] atomic fractions *WTFRAC* weight fractions Can be input as a vector with up to MT entries (one for each Material) [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13.] If less than MT entries are made, the last entry will be used to fill out the array to a length of MT.
ATWT [≤2*NISO] {required[b]}	Atomic weights of the isotopes. If using MATSPEC=ATFRAC or WTFRAC, atomic weights must be available to the code. Entries for the ATWT array are made in pairs, as follows: ATWT= $iso_1\ atwt_1\ iso_2\ atwt_2$ .... where iso_n is the isotope name (identifier) for isotope n on the cross-section library and $atwt_n$ is that isotope's atomic weight. [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13].

a. ATFRAC and WTFRAC cannot be used with PREMIX.

b. Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.

Block-V Details: Solver Input

Desired Calculation

Name	Comments
IEVT	Calculation type: Enter one of the following values: Value — Calculation Desired *0* — inhomogeneous source (default). *1* — k_{eff}. *2* — α (time absorption) search. *3* — concentration search. *4* — dimension search.
ISCT	Legendre order of scattering (default = 0).
ITH	*0/1* = direct/adjoint calculation (default = 0).
IBL	Left boundary condition[a]. *0/1/2/3* = vacuum/reflective/periodic/white (default = vacuum).
IBR	Right boundary condition. *0/1/2/3* = vacuum/reflective/periodic/white (default = vacuum).
IBT	Top boundary condition. *0/1/2/3* = vacuum/reflective/periodic/white (default = vacuum).
IBB	Bottom boundary condition. *0/1/2/3* = vacuum/reflective/periodic/white (default = vacuum).

a. The left boundary condition applies only for x,y geometry.

Iteration Controls

Name	Comments
EPSI	Convergence precision (default=0.0001).
IITL	Maximum number of inner iterations per group at first (default=1).
IITM	Maximum number of inners allowed when fission source is near convergence (default chosen by code).
OITM	Maximum number of outer iterations (default=20).
ITLIM	Number of seconds time limit (default=unlimited).
NOSIGF[a]	Inhibit fission multiplication in a fixed source problem. *0/1* = no/yes.

a. The situation envisioned by this option is that a fission problem has previously been run in which case a FIXSRC file will have been automatically written. That FIXSRC file will contain the pointwise source given by $(1/k_{eff})\,\chi_g\Phi$ where Φ is the fission distribution. Then the problem is rerun with NOSIGF=1 and INSORS=1 to achieve the same answer as the original. This can then be used to study differences from the original system that do not impact the fission source such as changes in the shield design of a nuclear reactor.

Acceleration Controls

Name	Comments
GREYACC	Upscatter acceleration[a]. *0/1* = *no/yes*[b].

a. See page 7-21 for details.

b. The user may also input the words *no* or *yes*.

K-Code Convergence {Optional}

Name	Comments
KCALC	Special Criticality Convergence Scheme. *0/1* = no/yes.

A special convergence scheme may be invoked for problems which require a good eigenvalue but do not require tight convergence of the pointwise fluxes. It consists of converging the eigenvalue but not the pointwise fluxes. Normally both must be converged. It also sets the default for eigenvalue convergence to 0.001 rather than 0.0001. To invoke this option to save running time, set the input parameter KCALC to unity.

Output Controls {Optional}

Name	Comments
FLUXP	Final flux print. *0/1/2* = no/isotropic/all moments.
XSECTP	Cross-section print. *0/1/2* = no/principal/all .
FISSRP	Fission source rate print. *0/1* = no/yes.
SOURCP	Source print. *0/1/2/3* = no/as input/normalized/both .
ANGP	Print the angular flux. *0/1* = no/yes. CAUTION! this is very LARGE output. ANGP=1 will also cause the RAFLXM or AAFLXM file to be written.
BALP	Coarse Mesh Interval Print Options. Enter one of the following values: Value — Description *0* — None *1* — Print coarse mesh balance tables only. *2* — Print coarse mesh negative flux fixup monitor only. *3* — Print both balance tables and flux fixup monitor.
RAFLUX	Prepare angular flux file (RAFLUX or AAFLUX). *0/1* = no/yes.
RMFLUX	Prepare flux moments file (RMFLUX or AMFLUX). *0/1* = no/yes.
ASRITE	0/i = No/Write right-going angular boundary flux from column i to file ARBFLUX.[a] The value of i must be between 1 and IT+1.
ASBOTT	0/j = No/Write down-going angular boundary flux from row j to file ARBFLUX. The value of j must be between 1 and JT+1.
ASTOP	0/j = No/Write up-going angular boundary flux from row j to file ARBFLUX. The value of j must be between 1 and JT+1.
ASLEFT	0/i = No/Write left-going angular boundary flux from column i to file ARBFLUX. The value of i must be between 1 and IT+1.

a. See "ARBFLUX" on page 15-17 for the format and order of the angular fluxes. The order of the data in the ARBFLUX file is compatible with the format required for inputting full angular boundary sources. For example, a right-going angular boundary flux specified by an ASRITE write may be used as a SALEFT source for a subsequent run by replacing the appropriate header card in the ARBFLUX file with "SALEFT=."

Miscellaneous Solver Input {Optional}

Name	Comments
TRCOR	Apply transport correction[a] to cross sections on MACRXS file. Enter one of the following words: Word — Description *DIAG* — Use diagonal transport correction. *BHS* — Use Bell-Hansen-Sandmeier correction. *CESARO* — Use Cesaro "correction." *NO* — (or omit entry) Use no correction.
NORM	Normalize the fission source rate to this value when IEVT≥1 or normalize the inhomogeneous source rate to this value when IEVT=0. NORM=0 means no normalization. (Integral of source rate over all angle, space, and energy = NORM, except for k_{eff} problems where the integral is equal to NORM*k_{eff}). Any fluxes printed by setting FLUXP nonzero will be normalized consistently with this source rate.
BHGT	Buckling height to use to correct for leakage. Units are centimeters if macroscopic cross section is in cm^{-1}. Ignore if r-z.
CHI [NGROUP;M]	Fission fraction born into each group.[b] Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.
DEN [IT;JT] or	Density factor to use at each fine mesh point. Applied to the zone macroscopic cross sections at each mesh interval.
DENX [IT][c] and/or	Density factor to use at each fine x-mesh (or r-mesh) (default=1).
DENY [JT]	Density factor to use at each fine y-mesh (or z-mesh) (or theta-mesh) (default=1).
WDAMP [NGROUP]	Flags to activate adaptive weighted diamond differencing (AWDD)[11] for each group. *0.0*/W = no/ activate AWDD with parameter W.[d] If W = 0.0, the default diamond with fixup is used.

a. For more information, see "Transport Corrections for the Cross Sections (TRCOR)" on page 7-31.

b. This input will override any previous CHI from earlier blocks or from any cross-section library which contains CHI.

c. In this second form, the density factor DEN(i,j), at mesh interval (i,j) is computed as follows:

DEN(i,j) = DENX(i)*DENY(j)

d. Recommend 1.0 < W < 4.0 for shielding applications.

Quadrature Details

Name	Description
GRPSN [NGROUP]	S_n order to be used for each group.
IQUAD	Source of quadrature constants. Enter one of the following: Value — Description *-3* — Get constants from SNCONS file. *-2* — Triangular Chebychev-Legendre built-in set. Any even value for ISN can be used up to 50. From 50 to 100, ISN must be in multiples of 10. See Ref. 9 for details. *1* — Traditional built-in constants. Any even value for ISN can be used between 2 and 16, inclusive. (This is the default). *2* — Rectangular Chebychev-Legendre built-in set. (REQUIRES ISN negative in Block-I!) Any even value for the absolute value of ISN can be used up to 50. From 50 to 100, the absolute value of ISN must be in multiples of 10. See Ref. 9 for details.
WGT[a] [MM[b]] {optional}	Quadrature weights.
MU [MM] {optional}	Mu cosines.
ETA [MM] {optional}	Eta cosines.

a. Presence of the WGT, MU, and ETA arrays overrides the IQUAD input.

b. MM=ISN*(ISN+2)/8 for IQUAD = -2,1.
MM = (ISN/2)**2 for IQUAD = 2.

Flux Guess From a File {Optional}

Name	Comments
INFLUX	Read the initial flux guess from a file.[a] *0/1* = no/yes. If ITH=0 and ISCT>0, and the flux moments file RMFLUX exists, read the initial flux guess from RMFLUX. Otherwise, read the initial flux guess from the RTFLUX file. If ITH=1 and ISCT>0, and the adjoint moments file AMFLUX exists, read the initial flux guess from AMFLUX. Otherwise, read the initial flux guess from the ATFLUX file.

a. There is presently no text input flux guess available for TWODANT.

General Eigenvalue Search Control[a] { IEVT >1}

Name	Comments
IPVT	Type of eigenvalue to search for in a concentration or dimension search. *0/1/2* = none / k_{eff} / α. (default = 1).
PV	Value of k_{eff} or α to which to search. (default = 1.0 if IPVT=1, 0.0 if IPVT=2.
EV	Initial search parameter. Value at which to start the search parameter. (default=0).
EVM	Initial search parameter increment. Amount by which to change search parameter in the first step of a search. (REQUIRED - there is no default).
XLAL	Lambda lower limit for search. (default = 0.01).
XLAH	Lambda upper limit for search. (default = 0.5).
XLAX	Lambda convergence criterion for second and subsequent search steps. (default = 10*EPSI).
POD	Parameter oscillation damper. (default=1.0).

a. See "Eigenvalue Searches" on page 7-33 for definitions of these quantities.

TWODANT can vary the composition or dimensions of a zone (or zones) in order to achieve a desired k-effective or alpha value. The search input consists of the above general search input plus input specific to the type of search being performed.

Dimension Search Input {Required if IEVT=4}

Name	Comments
XM [IM]	x-dimension fractional change per coarse mesh (or r).
YM [JM]	y-dimension fractional change per coarse mesh (or z) (or theta).

The dimension search requires the XM and/or YM input as well as the general search input above. During the search, TWODANT varies the search parameter (sometimes called the search eigenvalue) denoted by EV in the following expressions to change the coarse mesh boundaries:

$$XMESH_{i+1} = XMESH_i + \{XMESH_{i+1}\text{-}XMESH_i\} * [1.0+EV*XM_i], \quad i=1, \ldots, IM$$

$$YMESH_{j+1} = YMESH_j + \{YMESH_{j+1}\text{-}YMESH_j\} * [1.0+EV*YM_j], \quad j=1, \ldots, JM$$

Although they may seem a bit awkward at first, the user will find these expressions to be quite flexible. With proper choice of the XM_i, and YM_j values, the user can move any or all of the coarse mesh boundaries while allowing others to remain stationary. The quantities in { } in the above expressions are always formed from the original input values.

Concentration Search Input {Required if IEVT=3}

Name	Description
The solver input for a concentration search is to set IEVT = 3 (page 3-50) and input the general eigenvalue search controls. But you must also input the ASGMOD[a] array in Block-IV.	

a. A concentration search involves the mixing instructions. A discussion of the ASGMOD array is found in the mixing input description on page 3-47.

Volumetric Source Options {Optional}

Name	Comments
INSORS	Read source from interface file FIXSRC. *0/1* = no/yes.
----- For a text-input source, choose one of the following options:	
Option 1:	
SOURCE [NGROUP; NMQ]	Source spectrum for each of NMQ[a] moments. (Spatial distribution is assumed to be flat with value unity).
Option 2:	(input both arrays)
SOURCX [IT;NMQ][b]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z, or theta) spatial distribution for each moment. (Spectrum is assumed to be flat with value unity)
Option 3:	(input all three arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCX [IT;NMQ]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z, or theta) spatial distribution for each moment.
Option 4:	
SOURCF [IT;JT*NGROUP*NMQ]	Spatial distribution for each row, group, and moment.
Option 5:	(input both arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCF [IT; JT*NMQ]	Spatial distribution for each row and moment.

a. NMQ is not an input value but is computed from the number of strings read. NMQ must correspond exactly to the number of moments in a P_n expansion of the source. The number of moments is (n+1)(n+2)/2. n must be less than or equal to ISCT. See page 12-24 for more details.

b. Only in option 4 is the complete pointwise source array, SOURCF(i,j,g,m), given. In all other cases, it must be formed from the lower dimension arrays that are input. That calculation is done by forming the product of those arrays. Thus, in option 3, where the source spectrum, SOURCE(g,m), and the spatial distributions SOURCX(i,m), SOURCY(j,m), are given (for moment m), the full source at mesh point (i,j) in group g for moment m is calculated as follows:

SOURCF(i,j,g,m) = SOURCE(g,m)*SOURCX(i,m)*SOURCY(j,m)

Boundary Source Input {Optional}

Name	Comments
----- For a text-input source, choose one of the following options:	
Option 1: Isotropic Boundary Source.	
SILEFT [NGROUP;JT]	Isotropic source on the left side. (Spectrum at each y mesh interval) (or z) (or theta).
SIRITE [NGROUP;JT]	Isotropic source on the right side.
SIBOTT [NGROUP;IT]	Isotropic source on the bottom side.
SITOP [NGROUP;IT]	Isotropic source on the top side.
Option 2: Full Angular Boundary Source.[a]	
SALEFT [MM*2;[b]NGROUP*JT]	Angular flux on the left for each angle, group, and y (or z) (or theta) mesh interval.
SARITE [MM*2;NGROUP*JT]	Angular fluxes on the right side.
SABOTT [MM*2;NGROUP*IT]	Angular fluxes on the bottom side.
SATOP [MM*2;NGROUP*IT]	Angular fluxes on the top side.
Option 3: Boundary Source From Vectors.[a c d]	
BSLFTG [NGROUP] BSLFTY [JT] BSLFTA [MM*2]	Spectrum on left side. Spatial distribution on left side. Angular distribution on left side.
BSRITG [NGROUP] BSRITY [JT] BSRITA [MM*2]	Spectrum on right side. Spatial distribution on right side. Angular distribution on right side.
BSBOTG [NGROUP] BSBOTY [IT] BSBOTA [MM*2]	Spectrum on bottom side. Spatial distribution on bottom side. Angular distribution on bottom side.
BSTOPG [NGROUP] BSTOPY [IT] BSTOPA [MM*2]	Spectrum on top side. Spatial distribution on top side. Angular distribution on top side.

a. The order of the angles is identical to that used in the S_n Constants table in the output file. The order of the angular quadrants is: mu<0, eta<0; mu>0, eta<0; mu<0, eta>0; and mu>0, eta>0, where each angular boundary source requires two quadrants for specification.

b. See "Quadrature Details" on page 3-54 for value of MM.

c. The full angular source is constructed from a product of the vectors. For example, the source in angle m on the left side of mesh row j for group g is (the source construction on the other faces is analogous):

$$S(m,g_j) = BSLFTG(g)*BSLFTY(j)*BSLFTA(m)$$

d. Any vector not entered explicitly is defaulted to unity.

First Collision Source Input {Optional}

Name	Comments
FCSRC	Use First Collision Source Option. Enter one of the following words: Word — Description *ANA*[a] — Use analytic first collision source.[12] Cylindrical (r,z) geometry only. *RAY*[b] — Use the ray tracing first collision source, (x,y) or (r,z) geometry only. *NO* — (or omit entry) - don't use
FCNRAY	Number of ray tracings/batch to use with ray tracing first collision source. (default = 10000).
FCNTR	Number of batches (trials) to use with ray tracing first collision source. If FCNTR is negative, read the restart file UCFLUX, and perform |FCNTR| additional batches of ray traces (default = 25).
FCWCO	Weight cutoff for the ray tracing first collision source. When a ray has been traced a sufficient distance to attenuate its weight to less than wtmin * FCWCO, where wtmin is the minimum source starting weight, then that ray is terminated. Default entry is 0.0, which corresponds to a FCWCO value of EPSI**2.

a. If FCSRC= ANA, the location of the point source on the z-axis is set by using the SOURCY [JT] input vector. The point source is located at the lower left hand corner of the (1,js) mesh cell, where js is the j-index of the source cell. For example, if the point source is to be located along the z-axis at a distance defined by the lower edge of the j=15 fine mesh cell, one would enter:

```
SOURCY= 14R0  1  F0
```

to locate the point source for the calculation. The energy spectrum of the source is entered via the SOURCE input vector [NGROUP entries]. The SOURCX input vector is not used for this option.

b. The ray tracing first collision source may be used with any of the volumetric or boundary source input options.

Monte Carlo Option Controls[a] {Optional}

Name	Comments
MCOPT {required}	Controls the use of the Monte Carlo/S_n option. A value of 1 turns on the MC/S_n option, while 0 results in a standard S_n calculation. *0/1* = no/yes. The MC/S_n option is only valid for x-y and r-z geometries and inhomogeneous source (IEVT=0) calculations. (default=0).
MCREG [NGROUP]	A value of 1 denotes energy groups in which the MC/S_n method is to be used, while 0 identifies groups in which only S_n is to be used. (default=f1).
MCIBND [2]	Locates the left and right cells of the designated Monte Carlo region, where the numerical values represent an S_n fine mesh cell. A value of zero automatically sets the boundary to the leftmost or rightmost cell, respectively. (default=0 0). Only a single, rectangular Monte Carlo region is allowed.
MCJBND [2]	Locates the bottom and top cells of the designated Monte Carlo region as described above. (default=0 0).
MCBLT	Thickness of the Monte Carlo boundary layer in terms of mean free paths. Only a single value for the mean free path is used along each face of the Monte Carlo region. The calculated group-dependent Monte Carlo region boundaries are listed in the Monte Carlo Setup Information in the output. (default=1.0).
MCITS	The maximum number of super-outer iterations allowed on the common boundary fluxes/sources between the Monte Carlo and S_n regions. (default=8).
MCNHIS	The approximate number of histories per trial to be used in sampling the Monte Carlo source. (default=10000).
MCNTR	The number of trials (batches) to be used in sampling the Monte Carlo source. (default=25).
MCISEED	Resets the initial random seed. (default=0).
MCIPRNT	Print level option for the Monte Carlo output. Allowable values are *0/1/2*. (default=0). The larger values print out increasingly more detailed information about the S_n/MC interface fluxes and sources.

a. On page 12-40 of the methods chapter will be found a discussion of the S_n/MC hybrid method

Monte Carlo Biasing Input {Optional[a]}

Name	Comments
MCSB	Biasing parameter for distributed source. Biases angular distribution of source according to q(u)=C*exp(MCSB*u), where u is the direction cosine along the y (or z) axis.
MCWC [2]	Weight cutoff parameters w1 and w2. When the weight of a particle in proportion to its source weight (w/ws) becomes smaller than w2, Russian roulette is played, with a probability of survival w/ws * 1/w1. If the particle survives, it is assigned weight w1 * ws. w2 must be less than or equal to w1.
MCCMIMP [IM;JM]	Array of coarse mesh cell importances for splitting/ Russian roulette. When crossing into cell j from cell i, a particle is assigned the relative importance $R_{ij} = r_j/r_i$. Then, with probability R_{ij} - Int R_{ij}, Int R_{ij} + 1 particles are created; otherwise, Int R_{ij} are created. The weight of all split particles is w/R_{ij}. A cell importance of zero eliminates all particles originating in or entering that coarse mesh.
MCEGIMP [NGROUP]	Array of energy group importances for splitting/ Russian roulette. Splitting/Russian roulette works as described above for coarse mesh cell importances.
MCLVIMP [ISN/2]	Level biasing parameters. A value of 0 eliminates the sampling of boundary fluxes within that quadrature level along the S_n/MC interface, while 1 allows normal sampling. See page 12-46 in the section on TWODANT methods for more details.

a. MCOPT must equal one in order to turn on the MC/S_n option and allow use of these options.

Monte Carlo Source Input {Optional[a]}

Name	Comments
MCPTSRC [1/2]	Gives coordinates (in S_n dimensions) for a point source (x-y geometry) or ring source (r-z geometry). If there is only one entry, then it represents the boundary location for a point beam source (use in conjunction with MCBSLA, MCBSRA, MCBSBA, or MCBSTA). If there are two entries, the first entry is the x (or r) coordinate, the second entry is the y (or z) coordinate, and the angular distribution is assumed to be isotropic.
MCBSLA [2]	Gives polar angle and azimuthal angle for a left boundary beam source. The polar angle is measured from the +y axis and the azimuthal angle from the +x axis. Angles are in radians so MCBSLA= 1.5708, 0.0, is normal to the left surface. Use with MCPTSRC for a point beam source, or an S_n boundary source for a distributed beam source.
MCBSRA [2]	Gives polar angle and azimuthal angle for a right boundary beam source. The polar angle is measured from the +y axis and the azimuthal angle from the -x axis. Angles are in radians so MCBSLA= 1.5708, 0.0, is normal to the right surface. Use with MCPTSRC for a point beam source, or an S_n boundary source for a distributed beam source.
MCBSBA [2]	Gives polar angle and azimuthal angle for a bottom boundary beam source. The polar angle is measured from the +y axis and the azimuthal angle from the +x axis. Angles are in radians so MCBSLA= 0.0, 0.0, is normal to the bottom surface. Use with MCPTSRC for a point beam source, or an S_n boundary source for a distributed beam source.
MCBSTA [2]	Gives polar angle and azimuthal angle for a top boundary beam source. The polar angle is measured from the +y axis and the azimuthal angle from the +x axis. Angles are in radians so MCBSLA=3.1416, 0.0, is normal to the top surface. Use with MCPTSRC for a point beam source, or an S_n boundary source for a distributed beam source.

a. MCOPT must equal one in order to turn on the MC/S_n option and allow use of these options.

The Monte Carlo/S_n option works with all volumetric and boundary source options described on page 3-57 through page 3-60. In addition, the singular sources described above may also be used with the Monte Carlo/S_n option.

Note that in order to provide storage for arrays required by the Monte Carlo/S_n option, a volumetric source must always be entered, even if it is not used. Furthermore, the number of moments entered for the volumetric source must correspond to the value of ISCT used in the problem.

Block-VI Details: Edit Input*

Edit Spatial Specifications {Required[a]}

Name	Comments
PTED	Do edits by fine mesh. *0/1* = no/yes.
ZNED	Do edits by zone. *0/1* = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below).
POINTS[≤IT*JT] {optional}	Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points).
EDZONE [IT;JT] {optional}	Edit zone number for each fine mesh interval. USED ONLY IF ZNED=1. (default= SOLVER coarse mesh interval numbers, see ZONES array, Block-II on page 3-36).

a. Either PTED or ZNED or both must be unity in order to produce reaction rate edits.

* More details for the input for edits are given in chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

Reaction Rates from Cross Sections[a] {Optional[b]}

Name	Comments
EDXS [≤NEDT] {required[c]}	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 the character name of the cross-section type. See the table "Edit Cross-Section Types by Position and Name" on page 3-67 or "MENDF Library Edit Cross Sections" on page 3-74 for the available names. NEDT is the total number of edit cross-section types available from the input cross-section library. (default = all shown in the table) Note: The cross-section types specified in this array apply to any or all of the following edit forms: RESDNT, EDISOS, EDCONS, EDMATS.
RESDNT	Do edits using the resident macroscopic cross section at each point. *0/1* = no/yes.
EDISOS [≤NISO]	Character names of the isotopes to be used in forming Isotopic reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDCONS [≤NISO]	Character names of the isotopes to be used in forming resident Constituent (partial macroscopic) reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDMATS [≤MT]	Character names of materials to be used in forming Material (macroscopic) reaction rates. The ordinal number may alternately be used, but is not recommended. (default = none).
XDF[d] [IT] YDF [JT]	Fine mesh density factors for the x (or r) and y (or z) (or theta) directions, respectively. The density factor is used to multiply resident Constituent (see EDCONS), macroscopic (see MACRO), and resident macroscopic (see RESDNT) reaction rates only. (default= all values unity).

a. See chapter "RUNNING THE EDIT MODULE" starting on page 8-1 for further discussion.

b. But either something in this grouping or in the "Reaction Rates from User Response Functions" grouping must be input in order to produce reaction rate edits.

c. You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.

d. If density factors were used in SOLVER to modify the cross sections at each mesh interval, the same density factors must be provided here in the XDF and/or YDF arrays as well. The density factor at mesh interval (i,j) is computed as:

XDF(i)*YDF(j)

Edit Cross-Section Types by Position and Name

CROSS-SECTION INPUT VIA ISOTXS or GRUPXS			CROSS-SECTION INPUT VIA ASCII TEXT		
Type	EDIT Position	Name[a]	Type	EDIT Position	Name
chi	1	CHI.....	not used	1	CHI.....
nu-fission	2	NUSIGF..	nu-fission	2	NUSIGF..
total	3	TOTAL...	total	3	TOTAL...
absorption	4	ABS.....	absorption	4	ABS.....
(n,p)	5	N-PROT..	1[b]	5	EDIT1...[c]
(n,d)	6	N-DEUT..	2	6	EDIT2...
(n,t)	7	N-TRIT..	3	7	EDIT3...
(n,alpha)	8	N-ALPH..	.	.	.
(n,2n)	9	N-2N....	.	.	.
(n,gamma)	10	N-GAMM..	.	.	.
fission	11	N-FISS..	N=IHT-3	4+N	EDITN...
transport	12	TRNSPT..			

a. Names are eight characters. A period within a name in this table denotes a blank.

b. Denotes position in the cross-section table. All cross sections in 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 EDNAME array of Block-III.

Reaction Rates from User Response Functions {Optional[a]}

Name	Comments
RSFE [NGROUP;M] {required}	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 with NGROUP entries beginning with group 1.
RSFX [IT;M][b]	Response function x (or r) distribution for M functions.
RSFY [JT;M]	Response function y (or z) (or theta) distribution for M functions.
{optional}	The above data are entered as M strings of IT or JT entries beginning with mesh point 1. (default=1.0).
RSFNAM [M]	Character names for the user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM).

a. But either something in this grouping or in the "Reaction Rates from Cross Sections" grouping must be input in order to produce reaction rate edits.

b. The M-th response function at space point (i,j) and energy group g is computed as:

RSFX(i,m)*RSFY(j,m)*RSFE(g,m)

Energy Group Collapse Specifications {Optional}

Name	Comments
ICOLL [NBG]	Edit energy group collapsing option: Number of SOLVER energy groups in each EDIT broad group. The NBG entries must sum to NGROUP. (default = 1 energy group per EDIT broad group).
IGRPED	Print option on energy groups. Enter one of the following values:

Value	Description
0	Print energy group totals only
1	Print broad groups only
2	Print broad groups only (same as 1)
3	Print both broad groups and totals

Reaction Rate Summing {Optional}

Name	Comments
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 "Edit Cross-Section Types by Position and Name" on page 3-67). Each of 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. (for more detail, see "Response Function Summing Options" on page 8-13).
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 page 8-13 for more detail.

Mass Inventories {Optional}

Name	Comments
MASSED	Calculate and print mass inventories by zone. *0/1/2/3* = none/ solver zones/edit zones/both (default=1). This option is active only if atomic weights are present. See ATWT on page 3-49.

Power Normalization {Optional}

Name	Comments
POWER {required}	Normalize to POWER megawatts.[a] All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be normalized. Fluxes are normally not printed here in the EDIT module, although they may be extracted by using a unit response function. Any such fluxes will also be normalized to POWER. Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 3-53).
MEVPER {required}	MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power. For the power calculation, TWODANT needs to know which cross section is the fission cross section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or GRUPXS library that designation is automatically provided (See "Edit Cross-Section Types by Position and Name" on page 3-67). But if one uses an ASCII text library, either ODNINP or XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 3-41).

a. Note that this normalization is meaningless if you are using the results of an adjoint run.

Miscellaneous Edit Items {Optional}

Name	Comments
RZFLUX	Write the CCCC standard zone[a] flux file RZFLUX or AZFLUX. *0/1* = no/yes.
RZMFLX	Write the code-dependent zone[b] flux moments file RZMFLX or AZMFLUX. *0/1* = no/yes.
EDOUTF[c]	ASCII output files control. Enter one of the following values:

Value	Description
-3	Write both EDTOGX (without scalar fluxes) and EDTOUT files.
-2	Write EDTOGX file (without scalar fluxes).
0	Write neither file. (default)
1	Write EDTOUT file.
2	Write EDTOGX file (with scalar fluxes).
3	Write both EDTOGX (with scalar fluxes) and EDTOUT files.

BYVOLP	Printed point reaction rates will have been multiplied by the mesh volume. *0/1* = no/yes.
AJED[d]	Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. *0/1* = regular/adjoint.
FLUXONE	Flux override. *0/1* = no/yes. Replaces all the input fluxes by unity. Useful for seeing the cross sections used in cross-section edits. WARNING! Meaningful reaction rates cannot be obtained when this switch is on.

a. RZFLUX and AXFLUX are organized by solver zones.

b. RZMFLX and AZMFLX are organized by solver zones.

c. See "ASCII File Output Capabilities (the EDOUTF Parameter)" on page 8-15.

d. See "Adjoint Edits" on page 8-15.

Special Plot Linkage {Optional}

Name	Comments
PRPLTED	Write an ASCII file of the pointwise reaction rates to link to the TECPLOT© plotting package available commercially for a SUN workstation. *0/1/2/3* = print only/nothing/tecplot file/both print and tecplot file.

To exercise this option, the user must have set PTED=1. The code will calculate reaction rates at all the fine mesh intervals and any POINTS input will be ignored.

To link to the TECPLOT© code, the user chooses option 2 or 3. Separate ASCII files called rsp.dat and med.dat will be written for the response function and material edits, respectively. These files are in input form for the TECPLOT© preprocessor.

If option 0 (print only) is chosen, no TECPLOT© files will be written but the reaction rates will be printed. The format of this printout is organized in a two-dimensional way unlike the normal printout from the EDIT module.

MENDF Library Edit Cross Sections

Reaction Type	Name	Description
χ	CHI	fission spectrum
$\nu\sigma_f$	NUSIGF	effective nu-sigma-fission
σ_t	TOTAL	Total cross section
σ_a	ABS	absorption[a]
(n,n)	MEND1	elastic scattering
(n,n´)	MEND2	inelastic scattering
(n,2n)	MEND3	n,2n scattering
(n,3n)	MEND4	n,3n scattering
(n,γ)	MEND5	gamma production
(n,α)	MEND6	alpha production
(n,p)	MEND7	proton production
(n,f)	MEND8	direct fission
(n,n´)f	MEND9	second-chance fission
(n,2n)f	MEND10	third-chance fission
(n,F)	N-FISS	[(n,F) = (n,f) + (n,n´)f + (n,2n)f]
χ_p	MEND12	prompt fission spectrum (only for fissionable materials)
χ_t	MEND13	total fission spectrum (only for fissionable materials)

a. σ_a for group g is defined as $\sigma_a = \sigma_t - \sum_{g'} \sigma_{g \to g'}$

When using the Los Alamos MENDF5 cross-section library with the codes, there are numerous edit cross sections available for use in the Edit Module. Since these come from the MENDF file, they are called upon with special character names in the Edit Module as part of the EDXS input.

These names are defined in the table above.

REFERENCES

1. 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.
2. 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.
3. R. D. O'Dell, F. W. Brinkley, D. R. Marr, and R. E. Alcouffe, "Revised User's Manual for ONEDANT: A Code Package for One- Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory report LA-9184-M, Revised, (December 1989).
4. 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).
5. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," *Nucl. Sci. Eng.* **64**, 344 (1977).
6. R. E. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Multigroup Diffusion Equation," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 340-351.
7. R. S. Baker, W. F. Filippone, and R. E. Alcouffe, "The Multigroup and Radial Geometry Formulation of the Monte Carlo/S_n Response Matrix Method," *Nucl. Sci. Eng.* **105**, 184 (1990).
8. R. E. Alcouffe, F. W. Brinkley, D. R. Marr, and R. D. O'Dell, "User's Guide for TWODANT: A Code Package for Two-Dimension, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory manual LA-10049-M, Rev. 1, (October 1984).
9. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).
10. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693, (March 1967).
11. R. E. Alcouffe, "An Adaptive Weighted Diamond Differencing Method for Three-Dimensional XYZ Geometry," *Trans Am Nuc Soc.* **68**, Part A, 206 (1993).
12. R. E. Alcouffe, R. D. O'Dell, and F. W. Brinkley, Jr., "A First-Collision Source Method That Satisfies Discrete S_n Transport Balance," *Nucl. Sci. Eng.* **105**, 198 (1990).

APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard k_{eff} Calculation

Sample problem 1 is a two group calculation of the k-eigenvalue for an R-Z model of a sodium cooled fast reactor. The geometric model contains two zones: a cylindrical core surrounded by a reflector zone. The core consists of mixed plutonium-uranium oxide fuel in steel pins cooled by liquid sodium. The reflector consists of steel pins cooled with liquid sodium. Both of these zones are homogenized in our model. We desire a calculation using only P_0 scattering from a multigroup cross-section set with two groups.

We also ask for some reaction rates to be calculated from the fluxes produced by the eigenvalue calculation.

Sample Problem 1: Output Description

Selected items in the output listing are described here. We focus on items particular to a two dimensional calculation. For a more thorough description of output items common to both one and two dimensional calculations, the reader is referred to Appendix A in the chapter "ONEDANT USER'S GUIDE".

The first item provided in the output is a listing of the input lines. This is shown on page 3-79. After the Block-I input describing the problem as a whole and the Block-II input describing the spatial meshing and zone assignments by coarse mesh, we describe the cross-section library in Block-III. P_0 cross sections for each isotope are entered in the input stream after Block-III. These isotopes are subsequently mixed in Block-IV to form the materials STEEL, FUEL, and SODIUM. These materials are then assigned with appropriate volume fractions to the CORE and REFLECTOR zones which correspond to the numbers 1 and 2 in the ZONES array of Block-II. In Block-V we specify that this is a k-eigenvalue calculation (ievt=1) with vacuum boundary conditions on the right, bottom and top boundaries (note: for R-Z the left boundary condition, IBL, is set to 1 by the code, meaning reflection, and the defaults for the rest are vacuum). The boundary conditions are verified in the output on page 3-83 and page 3-85. The fission source is normalized to 1.0. We are asking to print the scalar flux, the fission distribution, and the macroscopic cross sections.

In the edit input (Block-VI), the code is asked to calculate reaction rate totals for each edit zone. The reaction rates desired are the default ones for cross sections, that is, CHI, NUSIGF, TOTAL, ABS, and EDIT1 for the resident material, as well as rates using the supplied response functions. EDIT1 is the default name for the first position in the cross-section library. Note the use of comments (using the slash, /) to organize and describe the input. The edit output is self explanatory and is listed starting on page 3-94.

As far as the rest of the output shown, most of the items are also self explanatory and give the same information as a ONEDANT problem except for the two-dimensional differences. However, we would like to focus a little more attention on two items that serve as diagnostics and goodness of solution verification for the problem run: the iteration monitor and the balance table.

In considering the iteration monitor (page 3-88), we recall that for eigenvalue problems we do source iteration for the inner or within-group scattering source and outer iterations for the fission source. We also recall that the default (and recommended) strategy to solve these problems is to do one inner iteration per group until the fission source has sufficiently converged, and then to fully converge the inner iterations on the flux. This is reflected in the monitor which is arranged in rows for each outer. The inner convergence is not shown until the fission source has converged to near the input convergence criterion. The first outer, designated outer 1, is a pure diffusion calculation; the rest include the transport sweeps with DSA acceleration. Each column of the monitor gives respectively, the current CPU time, the transport outer counter, the number of transport inners for this outer (usually equal to the number of groups until the source is converged), the number of multigroup DSA iterations (sub-outers), the eigenvalue estimate at this outer, the precision of the eigenvalue (change from the previous transport outer), the maximum pointwise flux change (not important until the source has converged), the maximum pointwise fission distribution error, and information on the status of the inners. Note that at outer 3 the source has sufficiently converged where now the inner iterations on the groupwise scalar flux can be carried to completion. Thus, it is seen that upon completion of outer iteration 4, the whole problem has been converged. From the first column, we see that this was done in 0.26 CPU seconds on the Cray YMP.

The next item is the balance table which gives quantities from the transport equation integrated over the entire spatial domain of the problem for each group and the sum of the groups. This particle balance is a measure of the integral goodness of the solution and is equal to 1.0-sources/losses. Thus, the extraneous source, the fission source, and the inscatter are the source ingredients. The outscatter, net leakage, and absorption are the losses. The table for this problem shows that balance is achieved to better than 1 part in 10^7 for each group and better than 1 part in 10^9 for the total or group summed balance. Another thing to notice in the group total row is the balance between inscatter and outscatter. This indicates how well the cross-section set is balanced in the scattering matrix since these two quantities should be the same in the absence of any (n,2n) or (n,3n) reactions. Treatment of the (n,2n) and (n,3n) reactions varies with the cross-section processor, but it is common practice to include the (n,2n) and (n,3n) cross sections in the absorption and total cross sections, thus getting the loss reaction rate correctly and then to include twice the (n,2n) cross section and thrice the (n,3n) cross section in the inscattering cross sections, thus getting the scattering source in the group correctly. In this case, the total inscatter should exceed the total outscatter by the amount of production due to the (n,2n) and (n,3n) reactions. The nprod spectrum gives the number of fission neutrons produced in each group and is useful for determining whether fissions are being produced in the thermal or fast range, for example.

Sample Problem 1: Output Listing

```
 generalized input module run on 11/15/94 with solver version 10-13-94beta---- release 2.5      machine d

                    ...listing of cards in the input stream...

  1.    2    0    0
  2.  sample problem for twodant user's guide
  3.  standard k calculation,  all input by means of card-images
  4.  /   geometry          - r,z
  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, and sodium
  9.  /                     - materials assigned to make zones named core
 10.  /                      and reflector
 11.  /   solver            - card input supplied
 12.  /   edits             - zone edits for resident materials
 13.  /
 14.  /
 15.  / * * * * block i * * * *
 16.     igeom=r-z, ngroup=2, isn=4 niso=7  mt=3 nzone=2  im=2 it=25
 17.     jm=3  jt=30   t
 18.  /
 19.  /
 20.  / * * * * block ii (geometry) * * * *
 21.     xmesh=0.0,30,45     xints=15,10   ymesh=0,15,60,75   yints=5,20,5
 22.     zones= 2r2;  1,2;  2r2      t
 23.  /
 24.  /
 25.  / * * * * block iii (cross sections) * * * *
 26.     lib= odnimp
 27.     maxord=0  ihm=6  iht=4  ihs=5  ifido=0  ititl=1
 28.     names= "o-16"  "na-23"  fe  cr  ni  "pu-239"  "u-238"
 29.  /
 30.  / ****** since lib=odnimp, 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  (o-16)     sample 2 group lmfbr cross sections
 37.        0.000       0.010       0.000       2.000       1.600       0.000 o16/1
 38.        0.000       0.000       0.000       3.600       3.600       0.390 o16/2
 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 group 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, fe .05, cr .016,  ni 0.01;
 65.            fuel  "pu-239" .0103,  "u-238" .0103    "o-16" .0412;
 66.            sodium  "na-23" .025
 67.     assign= core    fuel .35, sodium .4, steel .25;
 68.             reflec  sodium .7, steel .3              t
 69.  /
 70.  /
 71.  / * * * * block v (solver) * * * *
 72.     ievt=1   isct=0  ibr=0
 73.     norm=1     fluxp=1  xsectp=2  fissrp=1
 74.     chi=0.6,0.4; 0.7, 0.3     t
 75.  /
 76.  /
 77.  / * * * * block vi (edits) * * * *
 78.    znedt=1, resdnt=1, t / *** zone edit for resident materials
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                                    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 for twodant user's guide                                *
*                          *  standard k calculation,  all input by means of card-images             *
*                          *                                                                         *
*                          ***************************************************************************
************************
*key end   block i read*
************************
*****************************************************************************************************************************************

*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                          ...block i - controls and dimensions...
*
*****************************************************************************************************************************************
*
*
*                                                ...dimensions (array name = dimens)...
*
*                           7  igeom   6/7/9/11 x-y/r-z/triangles/r-theta
*                           2  ngroup  number of energy groups
*                           4  isn     angular quadrature order
*                           7  niso    number of input isotopes (from isotxs, grupxs, or cards)
*                           3  mt      number of permanent materials
*                           2  nzone   number of zones
*                           2  im      number of coarse mesh x intervals
*                          25  it      number of fine mesh intervals
*                           3  jm      number of coarse mesh y intervals
*                          30  jt      number of fine mesh y intervals
*
*                                                   ...storage...
*                                                  mxdlcm=  140000
*                                                  mxdscm=   40000
*
*****************************************************************************************************************************************
*******************************
*key end   block ii read-geom*
*******************************
*
*                        7  igeom    1/2/3/6/7/8/9/11/14
```

```
******************************
*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   -1/0/1/2 - dtf(4e18.0)/dtf/fixed fido/free fido library
*                            1  ititl   0/1 - no/yes there is a title card before each table
*                            0  i2lp1   0/1 - no/yes library higher order scattering contains 2l+1 factor
*                            0  savbxs  0/1 - no/yes save binary xslib file (filename=bxslib)
*                            1  kwikrd  0/1 - full fido read/quick fido read (default=quick)
*
*
*                                                  ...energy structure...
*
*  group      chi          vel       lower bound  upper bound  group      chi          vel       lower bound  upper bound
*  -----  -----------  -----------  -----------  -----------  -----  -----------  -----------  -----------  -----------
*     1   0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00     2   0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
*                                          last neutron group(lng) is number   2
*
*
*                            0  balxs   -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance
*
*
*                                                  ...edit position names...
*
*                                                                  card
*                                                  position edname position
*                                                  -------- ------ --------
*                                                      1    chi
*                                                      2    nusigf     3
*                                                      3    total      4
*                                                      4    abs        2
*                                                      5    edit1      1
******************************
*key start card libe read    *
******************************
*
*                                          ...header cards from the card library...
*
*                                isotope isotope
*                                number   name   order          header card
*                                -------  ------ -----   ---------------------
*                                   1.    o-16    p0   -   oxygen-16  (o-16)    sample 2 group lmfbr cross sections
*                                   2.    na-23   p0   -   sodium     (na-23)   sample 2 group lmfbr cross sections
*                                   3.    fe      p0   -   iron       (fe)      sample 2 group lmfbr cross sections
*                                   4.    cr      p0   -   chromium   (cr)      sample 2 group lmfbr cross sections
*                                   5.    ni      p0   -   nickel     (ni)      sample 2 group lmfbr cross sections
*                                   6.    pu-239  p0   -   plutonium  (pu-239)  sample 2 group lmfbr cross sections
*                                   7.    u-238   p0   -   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   1.03000E-02, u-238    1.03000E-02, o-16     4.12000E-02,
*       3. sodium    na-23    2.50000E-02,
*
*     assign
*     ------
*
*       1. core      fuel     3.50000E-01, sodium   4.00000E-01, steel    2.50000E-01,
*       2. reflec    sodium   7.00000E-01, steel    3.00000E-01,
************************
*key start mix card xs *
************************
************************
*key end   mix card xs *
************************
*********************************************************************************************************************************
*****************************
*key end   block v read-solvr*
*****************************
*****************************
*key end   block vi read-edit*
*****************************
************************
*key end   input module*
************************
*
```

```
****************************************************************************************************************************************************
****************************************************************************************************************************************************
                    this twodant problem run on 11/15/94 with solver version 10-13-94beta--- release 2.5      machine d
* sample problem for twodant user's guide
* standard k calculation,  all input by means of card-images
*
****************************************************************************************************************************************************
*
*                                        ...block v -- solver input...
*
****************************************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*                                        ...required input(array name = solin)...
*
*          1         1       ievt   0/1/2/3/4 - type of calculation
*                                      0 inhomogeneous source
*                                      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
*          0         0       ibt    0/1/2/3 - top boundary condition
*                                      vacuum/reflective/periodic/white
*          0         0       ibb    0/1/2/3 - bottom boundary condition
*                                      vacuum/reflective/periodic/white
*
*
*                                        ...convergence controls(array name = iter)...
*
*      0.000E+00 1.000E-04  epsi   inner iteration convergence criterion (default=0.0001)
*          0         1       iitl   maximum number of inner iterations per group until fission source is near
*                                   convergence, i.e. lambda is near convergence. (default=1)
*          0        30       iitm   maximum number of inner iterations per group when close to fission source convergence
*                                   (default calculated)
*          0        20       oitm   maximum number of outer iterations (default=20)
*          0         0       itlim  iteration time limit (seconds)
*
*
*                                        ...miscellaneous parameters...
*
*                   0       nosigf set fissions zero when source calculation (0/1 no/yes)
*                   0       asleft write in-going angular boundary flux to file arbflux (0/i no/yes)
*                   0       asrite write out-going angular boundary flux to file arbflux (0/i no/yes)
*                   0       astop  write up-going angular boundary flux to file arbflux (0/j no/yes)
*                   0       asbott write down-going angular boundary flux to file arbflux (0/j no/yes)
*                   0       iufcs  fcsrc option (0/1/2 no/analytic/ray tracing)
*
*
*
****************************************************************************************************************************************************
****************************************************************************************************************************************************


****************************************************************************************************************************************************
****************************************************************************************************************************************************
*
*                                    ...block v -- solver input (continued)...
*
****************************************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*
*                                        ...miscellaneous parameters(array name = misc)...
*
*      0.000E+00 0.000E+00  bhgt   buckling height
*      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     10000       fcmxy  number of rays/trial for fcsrc rt option
*          0        25       fcntr  number of trials for fcsrc rt option
*          0         1       iquad  -3/-2/1/2/3 - source of quadrature constants (default=1)
*                                     -3 sncon file
*                                     -2 hybrid product set (triangular arrangement)
*                                      1 old twotran built-in set
*                                      2 product set (rectangular arrangement)
*                                      3 card input
*
*
*                                          ...output controls(array name = solout)...
*
*                      1  fluxp  0/1/2   none/isotropic/all moments - flux print
*                      2  xsectp 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 inhomogeneous source
*                      0  angp   0/1     no/yes - print angular fluxes
*                      0  rafluxm 0/1    no/yes - write angular fluxes to file raflxm or aaflxm(if ith=1)
*                      0  rmflux 0/1     no/yes - write flux moments to file rmflux
*                      0  balp   0/1     no/yes - print coarse mesh balances
*
*
*                                          ...parameters inferred from input arrays...
*
*                                  2  inchi  0/1/2 none/one chi/zonewise chi
*                                  0  isdenx 0/1/n - none/x density vector/full matrix
*                                  0  isdeny 0/1 no/yes - use y density vector
*                                  0  iqan   source anisotropy
*                                  0  isorse number of source moments input
*                                  0  isorsx number of sourcx moments input
```

```
*                          0  isorsy number of sourcy moments input
*                          0  isorsf number of sourcf moments input
*                          0  iql    -1/0/1/2 isotropic/none/all angles/vectors -left boundary source
*                          0  iqr    -1/0/1/2 isotropic/none/all angles/vectors -right boundary source
*                          0  iqt    -1/0/1/2 isotropic/none/all angles/vectors -top boundary source
*                          0  iqb    -1/0/1/2 isotropic/none/all angles/vectors -bottom boundary source
*
*
*
**************************************************************************************************************************************
**************************************************************************************************************************************


**************************************************************************************************************************************
**************************************************************************************************************************************
*
*
*                                  ...parameters from block i...
*
*                        7  igeom   6/7/11  x-y/r-z/r-theta
*                        2  ngroup  number of energy groups
*                        4  isn     angular quadrature order
*                        3  mt      number of permanent materials
*                        2  nzone   number of zones
*                        2  im      number of coarse mesh x intervals
*                        3  jm      number of coarse mesh y intervals
*                       25  it      number of fine mesh x intervals
*                       30  jt      number of fine mesh y intervals
*
*
*
*
**************************************************************************************************************************************
**************************************************************************************************************************************
```

```
********************************************************************************************************************************
********************************************************************************************************************************
*
*
*                                        ...material assignments to zones...
*
*
******************************
*key start matls to zones    *
******************************
*
*                    zone cross section = sum over matls in the zone of (matl cross section)*( c0 + c1*cmod )
*                                            with
*                                         cmod =  0.000000E+00
*
*                              entry   zone      material        c0             c1
*                                    no. name    no. name
*
*                                1     1 core     2 fuel       3.500000E-01   0.000000E+00
*                                2     1 core     3 sodium     4.000000E-01   0.000000E+00
*                                3     1 core     1 steel      2.500000E-01   0.000000E+00
*                                4     2 reflec   3 sodium     7.000000E-01   0.000000E+00
*                                5     2 reflec   1 steel      3.000000E-01   0.000000E+00
*
*
*
************************
*key start storage map *
************************
*
*  scm storage summary...
*
*     total scm required for this problem       10997
*     maximum scm available (maxscm= )          40000
*     scm required for transport                 9464
*     scm required for diffusion                10978
*
*
*  lcm storage summary...
*
*     lcm selected for this problem                  12689
*     maximum lcm specified (maxlcm= )              140000
*          Storage required for all quantities  in core is   12689
*
*          diffusion params are in core.  If O.K. to put in core, set MAXLCM .ge.    12689
*          diffusion fluxes are in core.  If O.K. to put in core, set MAXLCM .ge.    10169
*          scalar tr fluxes are in core.  If O.K. to put in core, set MAXLCM .ge.     9363
*          Storage required for all quantities  on disk is    8283
*


      12689 words lcm required lcmadd
*
* zero flux
*
************************
*key start sn constants*
************************
*
* s 4 constants for groups  1 to  2
*                 mu            eta           weight
*
*     1      0.30163878E+00  0.90444905E+00  0.83333333E-01
*     2      0.90444905E+00  0.30163878E+00  0.83333333E-01
*     3      0.30163878E+00  0.30163878E+00  0.83333333E-01
*************************************************************zones map*********************************************************
************************
*key start zones map   *
************************
    z                                zones by coarse mesh, origin at lower left.
  row  m                             m is number of fine intervals per coarse mesh.
  75.0000  rvvvvvvvvvvv
           r     *     v
    3   5  r   2 *   2 v
           r     *     v
  60.0000  r***********v
           r     *     v
    2  20  r   1 *   2 v
           r     *     v
  15.0000  r***********v
           r     *     v
    1   5  r   2 *   2 v
           r     *     v
   0.0000  rvvvvvvvvvvv
     r     0.   30.   45.
         0000  0000  0000
      m       15    10
  column       1     2
```

```
...cross section related data from file macxs  11/15/11:01: version  1 ...

 1 steel     2 fuel      3 sodium

                                    ...cross sections for legendre orders up to p0...

*key start mac cross sections*

                                                  group  1

                                    ...principal cross sections...

                        zone          chi         nu*fission      total       absorption
                      no.  name
                       1  core      6.0000E-01    2.5848E-02    1.2414E-01    8.9879E-03
                       2  reflec    7.0000E-01    0.0000E+00    8.3710E-02    4.5740E-04

                                    ...scattering matrices...
                                      (21+1 not included)

   zone   order  first grp    cross sections
     1      0      1          9.0947E-02
     2      0      1          6.8070E-02

                                                  group  2

                                    ...principal cross sections...

                        zone          chi         nu*fission      total       absorption
                      no.  name
                       1  core      4.0000E-01    1.7304E-02    2.7829E-01    1.1145E-02
                       2  reflec    3.0000E-01    0.0000E+00    1.8550E-01    4.2350E-04

                                    ...scattering matrices...
                                      (21+1 not included)

   zone   order  first grp    cross sections
     1      0      2          2.6714E-01    2.4203E-02
     2      0      2          1.8508E-01    1.5183E-02
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                        ...iteration controls and criteria...
*
*****************************************************************************************************************************
*
*
*                                              ***iteration criteria***
*
*                                                    transport inners
*
*        criterion    quantity to test                                    value     action taken if value exceeded
*        ---------    ----------------                                    -----     ------------------------------
*           iitl   - inner iteration count until near lambda               1         terminates inners
*                    (i.e. fission source) convergence
*           iitm   - inner iteration count when near lambda               30         terminates inners
*                    (i.e. fission source) convergence
*           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
*        ---------    ----------------                                    -----     ------------------------------
*           oitmd  - sub-outer iteration count                            40         terminates sub-outers
*           eps    - diffusion lambda-1.0 (see note below)             1.00E-04     does another sub-outer
*           eps    - fractional ptwise fission change per sub-outer    1.00E-04     does another sub-outer
*                    (see note below)
*
*           note: eps, when the problem is finally converged, will equal epsi, the value shown above. however,
*                 early in the iteration process, a larger value may be used to avoid unnecessary iterations.
*
*                                                 final convergence criteria
*
*        criterion    quantity to test                                    value     action taken if value exceeded
*        ---------    ----------------                                    -----     ------------------------------
*           oitm   - outer iteration count                                20         quits with error message
*           epsi   - transport lambda-1.0                              1.00E-04     does another outer
*
*****************************************************************************************************************************
*****************************************************************************************************************************
```

```
*************************************************************************************************************************************
*************************************************************************************************************************************
*
*                    ...flux and eigenvalue convergence as monitored by twodant...
*
*************************************************************************************************************************************
*****************************
*key start iteration monitor *
*****************************
*
*
* cpu time outer       diffusion     k-eff                    max ptwise   max ptwise   inners
*  (sec)   no. inners sub-outers  eigenvalue   lambda-1      flux change  fiss change converged
*   3.81     0
*   3.86     1      0       4     0.94013130  6.55942E-03   0.00000E+00  1.38816E+00  **no**
*   3.91     2      2       2     0.98011716  4.05061E-02   4.57809E-01  4.71563E-03  **no**
*   3.97     3      2       2     0.98056516  2.63283E-04   3.79512E-02  5.66785E-04  **no**
*
*    ------------------------------------------------------------------
*
*            — inner iteration summary for outer iteration no.   4 —
*
*                              iter per max flux   at
*                            group  group   change  mesh
*                              1      3   0.90E-04  22,  7
*                              2      3   0.46E-04  16,  6
*
*
* cpu time outer       diffusion     k-eff                    max ptwise   max ptwise   inners
*  (sec)   no. inners sub-outers  eigenvalue   lambda-1      flux change  fiss change converged
*   4.07     4      6       1     0.98063032  5.95137E-05   8.95425E-05  8.82870E-05   yes
*
*              $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance = -6.39737E-10     total inners all outers =   10
*
*************************************************************************************************************************************


*************************************************************************************************************************************
*************************************************************************************************************************************
*
*                                        ...group edit and balances upon convergence...
*************************************************************************************************************************************
*
*
*                    ...title--- sample problem for twodant user's guide                              ...
*
*
*                                        ...system balance tables...(neutrons only)
*
*****************************
*key start balance table    *
*****************************
*
*
*       gp          source       fission source        in scatter        self scatter       out scatter        net leakage
*
*        1   0.0000000E+00        6.0000000E-01      -1.0658141E-14       1.4710377E+00      3.7518535E-01      1.1425647E-01
*        2   0.0000000E+00        4.0000000E-01       3.7518655E-01       1.5023756E+01      5.6843419E-14      3.3295966E-01
*
*      tot   0.0000000E+00        1.0000000E+00       3.7518655E-01       1.6494794E+01      3.7518535E-01      4.4721612E-01
*
*
*
*       gp       absorption     particle balance       right leakage  horizontal leakage       top leakage     vertical leakage
*
*        1   1.1055814E-01        7.0097627E-08       7.1156679E-02       7.1156679E-02      2.1549898E-02      4.3099788E-02
*        2   4.4222574E-01       -5.5391062E-08       2.0615163E-01       2.0615163E-01      6.3404025E-02      1.2680803E-01
*
*      tot   5.5278388E-01       -6.3973715E-10       2.7730830E-01       2.7730830E-01      8.4953923E-02      1.6990782E-01
*
*
*
*       gp      left leakage      bottom leakage      nprod spectrum
*
*        1   0.0000000E+00        2.1549890E-02       3.1959352E-01
*        2   0.0000000E+00        6.3404008E-02       6.8040648E-01
*
*      tot   0.0000000E+00        8.4953898E-02       1.0000000E+00
*
*    Timing for sweeper and DSA...tsweep=   0.105 and tdsa=   0.139 and tgrey=   0.000 seconds.
*
*
*************************************************************************************************************************************


integral summary information
summary integral-k-eff     9.8063032E-01
integral-source-i             neutron  0.0000000E+00
integral-fission-i            neutron  1.0000000E+00
integral-in-scak-i            neutron  3.7518655E-01
integral-self-scak-i          neutron  1.6494794E+01
integral-out-scak-i           neutron  3.7518535E-01
integral-net lkage-i          neutron  4.4721612E-01
integral-absorption-i         neutron  5.5278388E-01
integral-right lkage-i        neutron  2.7730830E-01
integral-horizontal lkage-i   neutron  2.7730830E-01
integral-top lkage-i          neutron  8.4953923E-02
integral-vertical lkage-i     neutron  1.6990782E-01
```

```
**********
**********
*
*                                   flux components for group    1
*
**********
******************
*key start fluxes*
******************
*
*
* component number  1
*
*         r mesh    1   r mesh    2   r mesh    3   r mesh    4   r mesh    5   r mesh    6   r mesh    7   r mesh    8
*
*    30 1.595029E-05 1.571272E-05 1.541475E-05 1.504720E-05 1.460919E-05 1.409374E-05 1.349629E-05 1.281232E-05
*    29 2.311136E-05 2.277000E-05 2.232458E-05 2.177882E-05 2.112780E-05 2.037238E-05 1.951197E-05 1.855876E-05
*    28 3.186912E-05 3.140297E-05 3.078132E-05 3.000014E-05 2.906186E-05 2.797446E-05 2.674161E-05 2.536675E-05
*    27 4.404188E-05 4.342482E-05 4.258809E-05 4.152235E-05 4.022141E-05 3.868273E-05 3.692088E-05 3.494758E-05
*    26 6.178798E-05 6.101562E-05 5.990594E-05 5.844219E-05 5.662550E-05 5.446248E-05 5.195565E-05 4.912686E-05
*    25 8.456315E-05 8.357625E-05 8.210590E-05 8.013195E-05 7.765163E-05 7.467449E-05 7.122664E-05 6.732493E-05
*    24 1.069893E-04 1.057820E-04 1.039419E-04 1.014505E-04 9.831162E-05 9.453317E-05 9.013650E-05 8.515655E-05
*    23 1.258438E-04 1.244429E-04 1.222968E-04 1.193700E-04 1.156572E-04 1.111867E-04 1.059808E-04 1.000706E-04
*    22 1.418799E-04 1.402857E-04 1.378555E-04 1.345443E-04 1.303442E-04 1.252643E-04 1.193375E-04 1.126204E-04
*    21 1.554780E-04 1.537285E-04 1.510482E-04 1.473902E-04 1.427562E-04 1.371492E-04 1.306285E-04 1.232076E-04
*    20 1.668036E-04 1.649250E-04 1.620358E-04 1.581068E-04 1.531042E-04 1.470749E-04 1.400583E-04 1.321129E-04
*    19 1.759303E-04 1.739405E-04 1.708868E-04 1.667332E-04 1.614756E-04 1.551348E-04 1.477214E-04 1.393503E-04
*    18 1.828347E-04 1.807762E-04 1.776201E-04 1.732945E-04 1.678476E-04 1.612398E-04 1.535401E-04 1.448195E-04
*    17 1.874937E-04 1.853923E-04 1.821530E-04 1.777396E-04 1.721227E-04 1.653360E-04 1.574250E-04 1.484646E-04
*    16 1.898730E-04 1.877217E-04 1.844335E-04 1.799492E-04 1.742544E-04 1.673957E-04 1.594091E-04 1.503147E-04
*    15 1.898730E-04 1.877217E-04 1.844335E-04 1.799492E-04 1.742544E-04 1.673957E-04 1.594091E-04 1.503147E-04
*    14 1.874937E-04 1.853923E-04 1.821529E-04 1.777395E-04 1.721227E-04 1.653360E-04 1.574250E-04 1.484646E-04
*    13 1.828347E-04 1.807762E-04 1.776200E-04 1.732944E-04 1.678476E-04 1.612397E-04 1.535400E-04 1.448195E-04
*    12 1.759303E-04 1.739404E-04 1.708868E-04 1.667332E-04 1.614755E-04 1.551347E-04 1.477214E-04 1.393502E-04
*    11 1.668035E-04 1.649249E-04 1.620357E-04 1.581067E-04 1.531042E-04 1.470749E-04 1.400583E-04 1.321128E-04
*    10 1.554779E-04 1.537284E-04 1.510481E-04 1.473901E-04 1.427561E-04 1.371491E-04 1.306285E-04 1.232075E-04
*     9 1.418798E-04 1.402856E-04 1.378554E-04 1.345442E-04 1.303441E-04 1.252642E-04 1.193375E-04 1.126203E-04
*     8 1.258437E-04 1.244428E-04 1.222967E-04 1.193699E-04 1.156571E-04 1.111866E-04 1.059807E-04 1.000705E-04
*     7 1.069892E-04 1.057819E-04 1.039418E-04 1.014505E-04 9.831155E-05 9.453311E-05 9.013645E-05 8.515651E-05
*     6 8.456308E-05 8.357618E-05 8.210584E-05 8.013189E-05 7.765157E-05 7.467444E-05 7.122660E-05 6.732490E-05
*     5 6.178794E-05 6.101558E-05 5.990590E-05 5.844215E-05 5.662546E-05 5.446245E-05 5.195562E-05 4.912683E-05
*     4 4.404186E-05 4.342480E-05 4.258806E-05 4.152233E-05 4.022139E-05 3.868271E-05 3.692087E-05 3.494756E-05
*     3 3.186910E-05 3.140295E-05 3.078130E-05 3.000012E-05 2.906184E-05 2.797444E-05 2.674160E-05 2.536674E-05
*     2 2.311134E-05 2.276998E-05 2.232456E-05 2.177881E-05 2.112779E-05 2.037237E-05 1.951196E-05 1.855875E-05
*     1 1.595027E-05 1.571271E-05 1.541474E-05 1.504718E-05 1.460918E-05 1.409373E-05 1.349629E-05 1.281231E-05
*
*         r mesh    9   r mesh   10   r mesh   11   r mesh   12   r mesh   13   r mesh   14   r mesh   15   r mesh   16
*
*    30 1.201907E-05 1.115349E-05 1.018631E-05 9.129910E-06 8.142366E-06 7.299048E-06 6.523604E-06 5.859663E-06
*    29 1.749443E-05 1.630896E-05 1.501179E-05 1.354301E-05 1.202260E-05 1.065230E-05 9.458964E-06 8.487987E-06
*    28 2.387514E-05 2.225342E-05 2.051696E-05 1.864385E-05 1.660522E-05 1.464335E-05 1.289514E-05 1.147999E-05
*    27 3.277193E-05 3.042269E-05 2.790225E-05 2.524819E-05 2.242422E-05 1.954041E-05 1.699830E-05 1.505027E-05
*    26 4.600094E-05 4.259406E-05 3.893287E-05 3.503205E-05 3.091898E-05 2.656081E-05 2.228423E-05 1.906251E-05
*    25 6.299440E-05 5.827823E-05 5.319225E-05 4.776973E-05 4.196846E-05 3.586246E-05 2.900541E-05 2.324139E-05
*    24 7.963704E-05 7.359140E-05 6.708028E-05 6.008305E-05 5.261316E-05 4.466004E-05 3.543138E-05 2.760247E-05
*    23 9.349081E-05 8.629368E-05 7.851823E-05 7.015106E-05 6.131568E-05 5.179908E-05 4.085821E-05 3.177064E-05
*    22 1.051273E-04 9.695113E-05 8.811923E-05 7.871360E-05 6.879420E-05 5.801535E-05 4.564173E-05 3.539368E-05
*    21 1.149891E-04 1.060438E-04 9.638335E-05 8.613290E-05 7.523362E-05 6.330173E-05 4.970421E-05 3.853168E-05
*    20 1.233290E-04 1.137135E-04 1.033490E-04 9.234047E-05 8.056401E-05 6.770251E-05 5.312209E-05 4.114813E-05
*    19 1.300649E-04 1.198793E-04 1.089537E-04 9.733551E-05 8.486603E-05 7.125773E-05 5.586901E-05 4.326463E-05
*    18 1.351099E-04 1.245520E-04 1.132374E-04 1.011197E-04 8.810039E-05 7.393675E-05 5.794961E-05 4.486439E-05
*    17 1.385359E-04 1.277273E-04 1.160948E-04 1.036386E-04 9.027667E-05 7.573637E-05 5.934620E-05 4.593657E-05
*    16 1.402393E-04 1.293036E-04 1.175351E-04 1.049136E-04 9.136348E-05 7.664036E-05 6.004761E-05 4.647662E-05
*    15 1.402393E-04 1.293036E-04 1.175351E-04 1.049136E-04 9.136348E-05 7.664035E-05 6.004761E-05 4.647662E-05
*    14 1.385358E-04 1.277272E-04 1.160947E-04 1.036386E-04 9.027666E-05 7.573637E-05 5.934619E-05 4.593657E-05
*    13 1.351099E-04 1.245520E-04 1.132374E-04 1.011197E-04 8.810038E-05 7.393674E-05 5.794960E-05 4.486439E-05
*    12 1.300648E-04 1.198793E-04 1.089537E-04 9.733550E-05 8.486601E-05 7.125771E-05 5.586901E-05 4.326463E-05
*    11 1.233290E-04 1.137135E-04 1.033489E-04 9.234046E-05 8.056400E-05 6.770250E-05 5.312208E-05 4.114813E-05
*    10 1.149891E-04 1.060437E-04 9.638333E-05 8.613289E-05 7.523361E-05 6.330173E-05 4.970421E-05 3.853168E-05
*     9 1.051273E-04 9.695110E-05 8.811921E-05 7.871359E-05 6.879419E-05 5.801535E-05 4.564173E-05 3.539368E-05
*     8 9.349077E-05 8.629365E-05 7.851821E-05 7.015105E-05 6.131568E-05 5.179909E-05 4.085821E-05 3.177064E-05
*     7 7.963701E-05 7.359138E-05 6.708027E-05 6.008304E-05 5.261317E-05 4.466005E-05 3.543138E-05 2.760248E-05
*     6 6.299438E-05 5.827821E-05 5.319224E-05 4.776973E-05 4.196846E-05 3.586246E-05 2.900542E-05 2.324139E-05
*     5 4.600092E-05 4.259405E-05 3.893286E-05 3.503205E-05 3.091897E-05 2.656081E-05 2.228423E-05 1.906251E-05
*     4 3.277191E-05 3.042268E-05 2.790224E-05 2.524818E-05 2.242421E-05 1.954041E-05 1.699830E-05 1.505027E-05
*     3 2.387513E-05 2.225341E-05 2.051695E-05 1.864384E-05 1.660521E-05 1.464334E-05 1.289513E-05 1.147999E-05
*     2 1.749442E-05 1.630896E-05 1.501178E-05 1.354300E-05 1.202260E-05 1.065229E-05 9.458958E-06 8.487982E-06
*     1 1.201906E-05 1.115348E-05 1.018631E-05 9.129905E-06 8.142361E-06 7.299043E-06 6.523600E-06 5.859658E-06
*
*         r mesh   17   r mesh   18   r mesh   19   r mesh   20   r mesh   21   r mesh   22   r mesh   23   r mesh   24
*
*    30 5.313602E-06 4.771480E-06 4.231251E-06 3.718878E-06 3.230698E-06 2.764216E-06 2.341843E-06 1.956440E-06
*    29 7.665094E-06 6.863381E-06 6.103877E-06 5.368055E-06 4.653634E-06 3.993242E-06 3.385315E-06 2.813322E-06
*    28 1.034476E-05 9.256340E-06 8.187615E-06 7.145179E-06 6.177154E-06 5.294319E-06 4.483773E-06 3.719634E-06
*    27 1.344971E-05 1.188733E-05 1.038960E-05 9.013195E-06 7.772648E-06 6.652385E-06 5.625309E-06 4.654474E-06
*    26 1.673728E-05 1.458734E-05 1.265024E-05 1.093042E-05 9.399940E-06 8.025103E-06 6.768123E-06 5.587001E-06
*    25 1.981156E-05 1.708871E-05 1.470204E-05 1.264073E-05 1.083111E-05 9.224488E-06 7.757684E-06 6.387923E-06
*    24 2.295434E-05 1.952007E-05 1.666863E-05 1.422519E-05 1.213560E-05 1.028156E-05 8.618757E-06 7.077756E-06
*    23 2.625461E-05 2.210859E-05 1.881082E-05 1.598777E-05 1.356255E-05 1.144966E-05 9.554815E-06 7.812777E-06
*    22 2.917923E-05 2.447294E-05 2.079273E-05 1.769983E-05 1.500913E-05 1.267293E-05 1.055812E-05 8.616276E-06
*    21 3.171622E-05 2.649979E-05 2.244564E-05 1.913587E-05 1.625646E-05 1.373322E-05 1.149934E-05 9.400505E-06
*    20 3.385862E-05 2.824350E-05 2.383559E-05 2.029990E-05 1.728210E-05 1.459614E-05 1.221932E-05 1.003742E-05
*    19 3.558498E-05 2.964904E-05 2.496832E-05 2.121014E-05 1.809142E-05 1.531886E-05 1.280141E-05 1.052375E-05
*    18 3.689048E-05 3.072397E-05 2.584010E-05 2.189343E-05 1.863602E-05 1.584509E-05 1.329032E-05 1.089781E-05
*    17 3.776757E-05 3.144235E-05 2.642770E-05 2.237328E-05 1.901858E-05 1.614772E-05 1.358540E-05 1.117242E-05
*    16 3.820813E-05 3.180569E-05 2.672494E-05 2.260463E-05 1.920022E-05 1.631737E-05 1.374953E-05 1.130725E-05
*    15 3.820813E-05 3.180569E-05 2.672494E-05 2.260462E-05 1.920021E-05 1.631737E-05 1.374953E-05 1.130725E-05
*    14 3.776757E-05 3.144234E-05 2.642770E-05 2.237328E-05 1.901857E-05 1.614772E-05 1.358540E-05 1.117242E-05
*    13 3.689047E-05 3.072397E-05 2.584010E-05 2.189343E-05 1.863602E-05 1.584508E-05 1.329032E-05 1.089781E-05
*    12 3.558498E-05 2.964903E-05 2.496831E-05 2.121013E-05 1.809142E-05 1.531886E-05 1.280140E-05 1.052375E-05
*    11 3.385862E-05 2.824350E-05 2.383559E-05 2.029990E-05 1.728210E-05 1.459614E-05 1.221932E-05 1.003742E-05
*    10 3.171622E-05 2.649979E-05 2.244564E-05 1.913587E-05 1.625646E-05 1.373322E-05 1.149934E-05 9.400505E-06
*     9 2.917923E-05 2.447294E-05 2.079273E-05 1.769983E-05 1.500913E-05 1.267294E-05 1.055812E-05 8.616278E-06
*     8 2.625461E-05 2.210859E-05 1.881082E-05 1.598777E-05 1.356256E-05 1.144966E-05 9.554819E-06 7.812779E-06
*     7 2.295434E-05 1.952007E-05 1.666864E-05 1.422520E-05 1.213561E-05 1.028156E-05 8.618761E-06 7.077760E-06
*     6 1.981156E-05 1.708872E-05 1.470204E-05 1.264074E-05 1.083112E-05 9.224493E-06 7.757688E-06 6.387926E-06
*     5 1.673728E-05 1.458734E-05 1.265025E-05 1.093042E-05 9.399943E-06 8.025106E-06 6.768126E-06 5.587004E-06
*     4 1.344971E-05 1.188733E-05 1.038960E-05 9.013195E-06 7.772648E-06 6.652386E-06 5.625310E-06 4.654474E-06
*     3 1.034476E-05 9.256336E-06 8.187611E-06 7.145176E-06 6.177151E-06 5.294317E-06 4.483771E-06 3.719632E-06
```

```
*     2 7.665089E-06 6.863377E-06 6.103872E-06 5.368050E-06 4.653629E-06 3.993238E-06 3.385311E-06 2.813319E-06
*     1 5.313598E-06 4.771476E-06 4.231247E-06 3.718874E-06 3.230694E-06 2.764213E-06 2.341840E-06 1.956438E-06
*
*         r mesh  25
*
*    30 1.583627E-06
*    29 2.251514E-06
*    28 2.964954E-06
*    27 3.703850E-06
*    26 4.435856E-06
*    25 5.060573E-06
*    24 5.592986E-06
*    23 6.149175E-06
*    22 6.768029E-06
*    21 7.391177E-06
*    20 7.911982E-06
*    19 8.311282E-06
*    18 8.616728E-06
*    17 8.810178E-06
*    16 8.892663E-06
*    15 8.892663E-06
*    14 8.810178E-06
*    13 8.616728E-06
*    12 8.311281E-06
*    11 7.911982E-06
*    10 7.391177E-06
*     9 6.768030E-06
*     8 6.149176E-06
*     7 5.592988E-06
*     6 5.060575E-06
*     5 4.435858E-06
*     4 3.703851E-06
*     3 2.964953E-06
*     2 2.251512E-06
*     1 1.583624E-06
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                    flux components for group    2
*
*********************************************************************************************************************************
*
*
* component number  1
*
*         r mesh   1   r mesh   2   r mesh   3   r mesh   4   r mesh   5   r mesh   6   r mesh   7   r mesh   8
*
*    30 6.260015E-05 6.199378E-05 6.104895E-05 5.976326E-05 5.813139E-05 5.616878E-05 5.389803E-05 5.132236E-05
*    29 1.068511E-04 1.058521E-04 1.042544E-04 1.020453E-04 9.924100E-05 9.585657E-05 9.193999E-05 8.753689E-05
*    28 1.487918E-04 1.473998E-04 1.451625E-04 1.420638E-04 1.381107E-04 1.333447E-04 1.278077E-04 1.215783E-04
*    27 1.927269E-04 1.909353E-04 1.880340E-04 1.839869E-04 1.788060E-04 1.725262E-04 1.652226E-04 1.569790E-04
*    26 2.402751E-04 2.380953E-04 2.344759E-04 2.293735E-04 2.228102E-04 2.148445E-04 2.055545E-04 1.950525E-04
*    25 2.929165E-04 2.902911E-04 2.858411E-04 2.795377E-04 2.714102E-04 2.615264E-04 2.499951E-04 2.369395E-04
*    24 3.469447E-04 3.438574E-04 3.385541E-04 3.310053E-04 3.212573E-04 3.093903E-04 2.955246E-04 2.798058E-04
*    23 3.967848E-04 3.932506E-04 3.871589E-04 3.784675E-04 3.672151E-04 3.535085E-04 3.374783E-04 3.192920E-04
*    22 4.416981E-04 4.377383E-04 4.309165E-04 4.211776E-04 4.085706E-04 3.931970E-04 3.751990E-04 3.547884E-04
*    21 4.812493E-04 4.769195E-04 4.694516E-04 4.587786E-04 4.449680E-04 4.281119E-04 4.084005E-04 3.860014E-04
*    20 5.149573E-04 5.103105E-04 5.022850E-04 4.908303E-04 4.759805E-04 4.578748E-04 4.368854E-04 4.126267E-04
*    19 5.424219E-04 5.375112E-04 5.290369E-04 5.169334E-04 5.012517E-04 4.821355E-04 4.597422E-04 4.343346E-04
*    18 5.633120E-04 5.582022E-04 5.493938E-04 5.367910E-04 5.204871E-04 5.005803E-04 4.772892E-04 4.508450E-04
*    17 5.773805E-04 5.721448E-04 5.630955E-04 5.501775E-04 5.334320E-04 5.130069E-04 4.890937E-04 4.619493E-04
*    16 5.844662E-04 5.791554E-04 5.699944E-04 5.569023E-04 5.399424E-04 5.192551E-04 4.950427E-04 4.675433E-04
*    15 5.844662E-04 5.791554E-04 5.699944E-04 5.569023E-04 5.399424E-04 5.192551E-04 4.950427E-04 4.675433E-04
*    14 5.773805E-04 5.721447E-04 5.630954E-04 5.501775E-04 5.334320E-04 5.130069E-04 4.890936E-04 4.619492E-04
*    13 5.633118E-04 5.582021E-04 5.493937E-04 5.367909E-04 5.204870E-04 5.005802E-04 4.772891E-04 4.508449E-04
*    12 5.424218E-04 5.375110E-04 5.290368E-04 5.169333E-04 5.012516E-04 4.821353E-04 4.597421E-04 4.343345E-04
*    11 5.149572E-04 5.103103E-04 5.022848E-04 4.908302E-04 4.759803E-04 4.578746E-04 4.368852E-04 4.126266E-04
*    10 4.812490E-04 4.769193E-04 4.694514E-04 4.587784E-04 4.449678E-04 4.281117E-04 4.084004E-04 3.860013E-04
*     9 4.416979E-04 4.377381E-04 4.309163E-04 4.211774E-04 4.085704E-04 3.931969E-04 3.751989E-04 3.547882E-04
*     8 3.967845E-04 3.932504E-04 3.871586E-04 3.784673E-04 3.672149E-04 3.535083E-04 3.374781E-04 3.192919E-04
*     7 3.469445E-04 3.438572E-04 3.385539E-04 3.310051E-04 3.212571E-04 3.093902E-04 2.955244E-04 2.798057E-04
*     6 2.929163E-04 2.902909E-04 2.858409E-04 2.795376E-04 2.714101E-04 2.615263E-04 2.499950E-04 2.369394E-04
*     5 2.402750E-04 2.380951E-04 2.344758E-04 2.293734E-04 2.228101E-04 2.148445E-04 2.055544E-04 1.950524E-04
*     4 1.927268E-04 1.909352E-04 1.880339E-04 1.839869E-04 1.788059E-04 1.725261E-04 1.652226E-04 1.569789E-04
*     3 1.487918E-04 1.473997E-04 1.451625E-04 1.420637E-04 1.381106E-04 1.333447E-04 1.278077E-04 1.215783E-04
*     2 1.068510E-04 1.058521E-04 1.042544E-04 1.020453E-04 9.924096E-05 9.585654E-05 9.193996E-05 8.753686E-05
*     1 6.260013E-05 6.199375E-05 6.104892E-05 5.976324E-05 5.813137E-05 5.616876E-05 5.389801E-05 5.132235E-05
*
*         r mesh   9   r mesh  10   r mesh  11   r mesh  12   r mesh  13   r mesh  14   r mesh  15   r mesh  16
*
*    30 4.850260E-05 4.544494E-05 4.218729E-05 3.884551E-05 3.548378E-05 3.213232E-05 2.878776E-05 2.589834E-05
*    29 8.268819E-05 7.748367E-05 7.193923E-05 6.616102E-05 6.031763E-05 5.448791E-05 4.874795E-05 4.378888E-05
*    28 1.147350E-04 1.073749E-04 9.961403E-05 9.151144E-05 8.328628E-05 7.512461E-05 6.704837E-05 6.014264E-05
*    27 1.479060E-04 1.381240E-04 1.277915E-04 1.170497E-04 1.060704E-04 9.525780E-05 8.481512E-05 7.585898E-05
*    26 1.834689E-04 1.709501E-04 1.576618E-04 1.438060E-04 1.295601E-04 1.152863E-04 1.016196E-04 9.067519E-05
*    25 2.225053E-04 2.068813E-04 1.902347E-04 1.728076E-04 1.547453E-04 1.364459E-04 1.180910E-04 1.040754E-04
*    24 2.624220E-04 2.435384E-04 2.234018E-04 2.022049E-04 1.801458E-04 1.576126E-04 1.344381E-04 1.167309E-04
*    23 2.991438E-04 2.772457E-04 2.538540E-04 2.291411E-04 2.034900E-04 1.771464E-04 1.500576E-04 1.294773E-04
*    22 3.321385E-04 3.075334E-04 2.812059E-04 2.534590E-04 2.246848E-04 1.950171E-04 1.644770E-04 1.413354E-04
*    21 3.611827E-04 3.342185E-04 3.053610E-04 2.749958E-04 2.434092E-04 2.107654E-04 1.772497E-04 1.519118E-04
*    20 3.859750E-04 3.569861E-04 3.259887E-04 2.933325E-04 2.593267E-04 2.242417E-04 1.882458E-04 1.610498E-04
*    19 4.061726E-04 3.755295E-04 3.427732E-04 3.082716E-04 2.723468E-04 2.352613E-04 1.972399E-04 1.685452E-04
*    18 4.215106E-04 3.896369E-04 3.555737E-04 3.196519E-04 2.822364E-04 2.436552E-04 2.041167E-04 1.742877E-04
*    17 4.318638E-04 3.991561E-04 3.641730E-04 3.273004E-04 2.889155E-04 2.493188E-04 2.087554E-04 1.781618E-04
*    16 4.370608E-04 4.039310E-04 3.685072E-04 3.311587E-04 2.922685E-04 2.521699E-04 2.110941E-04 1.801190E-04
*    15 4.370608E-04 4.039310E-04 3.685072E-04 3.311587E-04 2.922685E-04 2.521699E-04 2.110941E-04 1.801190E-04
*    14 4.318637E-04 3.991561E-04 3.641730E-04 3.273004E-04 2.889155E-04 2.493187E-04 2.087554E-04 1.781617E-04
*    13 4.215105E-04 3.896369E-04 3.555737E-04 3.196519E-04 2.822363E-04 2.436551E-04 2.041167E-04 1.742876E-04
*    12 4.061725E-04 3.755295E-04 3.427732E-04 3.082716E-04 2.723467E-04 2.352613E-04 1.972399E-04 1.685452E-04
*    11 3.859749E-04 3.569860E-04 3.259886E-04 2.933325E-04 2.593267E-04 2.242417E-04 1.882458E-04 1.610498E-04
*    10 3.611826E-04 3.342184E-04 3.053610E-04 2.749958E-04 2.434092E-04 2.107653E-04 1.772497E-04 1.519118E-04
*     9 3.321384E-04 3.075334E-04 2.812058E-04 2.534589E-04 2.246848E-04 1.950171E-04 1.644770E-04 1.413354E-04
*     8 2.991437E-04 2.772457E-04 2.538540E-04 2.291411E-04 2.034900E-04 1.771464E-04 1.500576E-04 1.294773E-04
*     7 2.624219E-04 2.435384E-04 2.234017E-04 2.022048E-04 1.801458E-04 1.576126E-04 1.344381E-04 1.167309E-04
*     6 2.225052E-04 2.068813E-04 1.902346E-04 1.728075E-04 1.547453E-04 1.364458E-04 1.180910E-04 1.040754E-04
*     5 1.834689E-04 1.709500E-04 1.576617E-04 1.438060E-04 1.295601E-04 1.152863E-04 1.016196E-04 9.067519E-05
*     4 1.479060E-04 1.381240E-04 1.277915E-04 1.170497E-04 1.060704E-04 9.525778E-05 8.481511E-05 7.585898E-05
*     3 1.147350E-04 1.073748E-04 9.961400E-05 9.151142E-05 8.328626E-05 7.512460E-05 6.704836E-05 6.014263E-05
*     2 8.268817E-05 7.748364E-05 7.193921E-05 6.616100E-05 6.031762E-05 5.448790E-05 4.874794E-05 4.378887E-05
```

```
*     1 4.850259E-05 4.544492E-05 4.218728E-05 3.884550E-05 3.548377E-05 3.213231E-05 2.878776E-05 2.589833E-05
*
*         r mesh  17  r mesh  18  r mesh  19  r mesh  20  r mesh  21  r mesh  22  r mesh  23  r mesh  24
*
*    30 2.346447E-05 2.105103E-05 1.870185E-05 1.646872E-05 1.426729E-05 1.203972E-05 9.839980E-06 7.635065E-06
*    29 3.961195E-05 3.556823E-05 3.170611E-05 2.792252E-05 2.410219E-05 2.032106E-05 1.657350E-05 1.277739E-05
*    28 5.441116E-05 4.889988E-05 4.355252E-05 3.819271E-05 3.290841E-05 2.773987E-05 2.262063E-05 1.742349E-05
*    27 6.852108E-05 6.158039E-05 5.471924E-05 4.795108E-05 4.136301E-05 3.489868E-05 2.848592E-05 2.193956E-05
*    26 8.224255E-05 7.397972E-05 6.572630E-05 5.764720E-05 4.973863E-05 4.197329E-05 3.423843E-05 2.634380E-05
*    25 9.456524E-05 8.498818E-05 7.544106E-05 6.611281E-05 5.700433E-05 4.806256E-05 3.917286E-05 3.013403E-05
*    24 1.053424E-04 9.440570E-05 8.366319E-05 7.321638E-05 6.306607E-05 5.311924E-05 4.327094E-05 3.325640E-05
*    23 1.161713E-04 1.036453E-04 9.167009E-05 8.006929E-05 6.886860E-05 5.795368E-05 4.716136E-05 3.623212E-05
*    22 1.264766E-04 1.124953E-04 9.927416E-05 8.658727E-05 7.434616E-05 6.250256E-05 5.082786E-05 3.902067E-05
*    21 1.356654E-04 1.204539E-04 1.061366E-04 9.250763E-05 7.937967E-05 6.666214E-05 5.418259E-05 4.156384E-05
*    20 1.436493E-04 1.273490E-04 1.120515E-04 9.757966E-05 8.371218E-05 7.027433E-05 5.710060E-05 4.381218E-05
*    19 1.502103E-04 1.330660E-04 1.169741E-04 1.017597E-04 8.725677E-05 7.324606E-05 5.949180E-05 4.564094E-05
*    18 1.552341E-04 1.374249E-04 1.207422E-04 1.049994E-04 8.999343E-05 7.552385E-05 6.133269E-05 4.703432E-05
*    17 1.586377E-04 1.403969E-04 1.233022E-04 1.071706E-04 9.182751E-05 7.706440E-05 6.259753E-05 4.800097E-05
*    16 1.603505E-04 1.418857E-04 1.245938E-04 1.082843E-04 9.276529E-05 7.783400E-05 6.321822E-05 4.848259E-05
*    15 1.603505E-04 1.418857E-04 1.245938E-04 1.082843E-04 9.276529E-05 7.783400E-05 6.321822E-05 4.848259E-05
*    14 1.586377E-04 1.403969E-04 1.233022E-04 1.071706E-04 9.182750E-05 7.706439E-05 6.259752E-05 4.800097E-05
*    13 1.552341E-04 1.374249E-04 1.207422E-04 1.049994E-04 8.999342E-05 7.552385E-05 6.133269E-05 4.703432E-05
*    12 1.502102E-04 1.330660E-04 1.169741E-04 1.017597E-04 8.725676E-05 7.324605E-05 5.949180E-05 4.564093E-05
*    11 1.436493E-04 1.273490E-04 1.120515E-04 9.757965E-05 8.371217E-05 7.027433E-05 5.710060E-05 4.381217E-05
*    10 1.356654E-04 1.204539E-04 1.061366E-04 9.250762E-05 7.937966E-05 6.666213E-05 5.418259E-05 4.156384E-05
*     9 1.264766E-04 1.124953E-04 9.927416E-05 8.658727E-05 7.434616E-05 6.250256E-05 5.082786E-05 3.902067E-05
*     8 1.161713E-04 1.036453E-04 9.167009E-05 8.006929E-05 6.886860E-05 5.795368E-05 4.716136E-05 3.623212E-05
*     7 1.053424E-04 9.440570E-05 8.366319E-05 7.321638E-05 6.306608E-05 5.311925E-05 4.327094E-05 3.325640E-05
*     6 9.456524E-05 8.498818E-05 7.544107E-05 6.611281E-05 5.700433E-05 4.806256E-05 3.917286E-05 3.013404E-05
*     5 8.224255E-05 7.397972E-05 6.572630E-05 5.764720E-05 4.973864E-05 4.197329E-05 3.423843E-05 2.634381E-05
*     4 6.852107E-05 6.158038E-05 5.471923E-05 4.795108E-05 4.136301E-05 3.489868E-05 2.848592E-05 2.193956E-05
*     3 5.441116E-05 4.889988E-05 4.355252E-05 3.819271E-05 3.290841E-05 2.773987E-05 2.262062E-05 1.742348E-05
*     2 3.961194E-05 3.556822E-05 3.170610E-05 2.792251E-05 2.410219E-05 2.032106E-05 1.657350E-05 1.277739E-05
*     1 2.346446E-05 2.105103E-05 1.870184E-05 1.646872E-05 1.426729E-05 1.203972E-05 9.839978E-06 7.635063E-06
*
*         r mesh  25
*
*    30 5.308129E-06
*    29 8.711050E-06
*    28 1.187033E-05
*    27 1.493643E-05
*    26 1.792672E-05
*    25 2.049668E-05
*    24 2.260386E-05
*    23 2.461695E-05
*    22 2.649252E-05
*    21 2.820936E-05
*    20 2.971610E-05
*    19 3.096339E-05
*    18 3.191268E-05
*    17 3.254939E-05
*    16 3.287524E-05
*    15 3.287524E-05
*    14 3.254939E-05
*    13 3.191268E-05
*    12 3.096338E-05
*    11 2.971610E-05
*    10 2.820936E-05
*     9 2.649252E-05
*     8 2.461695E-05
*     7 2.260386E-05
*     6 2.049668E-05
*     5 1.792672E-05
*     4 1.493643E-05
*     3 1.187033E-05
*     2 8.711048E-06
*     1 5.308128E-06

   interface file fixsrc written with the fission source.
********************************************************************************************************************************************
********************************************************************************************************************************************
*
*                                                    fission neutron distribution
*
********************************************************************************************************************************************
*
*
******************************
*key start fission source    *
******************************
*
*
* component number  1
*
*         r mesh  1   r mesh  2   r mesh  3   r mesh  4   r mesh  5   r mesh  6   r mesh  7   r mesh  8
*
*    30 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    29 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    28 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    27 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    26 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    25 7.254403E-06 7.183464E-06 7.068455E-06 6.908359E-06 6.703610E-06 6.455628E-06 6.166971E-06 5.840206E-06
*    24 8.768974E-06 8.684344E-06 8.545013E-06 8.349994E-06 8.100181E-06 7.797169E-06 7.443592E-06 7.042873E-06
*    23 1.011876E-05 1.002139E-05 9.860505E-06 9.634460E-06 9.343779E-06 8.991047E-06 8.579099E-06 8.111638E-06
*    22 1.131043E-05 1.120071E-05 1.101985E-05 1.076574E-05 1.043902E-05 1.004169E-05 9.577062E-06 9.050252E-06
*    21 1.234631E-05 1.222617E-05 1.202766E-05 1.174842E-05 1.138967E-05 1.095306E-05 1.044343E-05 9.864020E-06
*    20 1.322234E-05 1.309337E-05 1.287982E-05 1.258005E-05 1.219378E-05 1.172464E-05 1.117661E-05 1.055493E-05
*    19 1.393349E-05 1.379708E-05 1.357151E-05 1.325471E-05 1.284746E-05 1.235277E-05 1.177366E-05 1.111763E-05
*    18 1.447343E-05 1.433181E-05 1.409781E-05 1.376792E-05 1.334501E-05 1.282974E-05 1.222769E-05 1.154469E-05
*    17 1.483730E-05 1.469239E-05 1.445207E-05 1.411446E-05 1.367951E-05 1.315065E-05 1.253237E-05 1.183106E-05
*    16 1.502141E-05 1.487391E-05 1.463039E-05 1.428794E-05 1.384726E-05 1.331201E-05 1.268660E-05 1.197568E-05
*    15 1.502141E-05 1.487391E-05 1.463039E-05 1.428794E-05 1.384726E-05 1.331201E-05 1.268660E-05 1.197568E-05
*    14 1.483730E-05 1.469238E-05 1.445207E-05 1.411446E-05 1.367951E-05 1.315065E-05 1.253237E-05 1.183106E-05
*    13 1.447343E-05 1.433180E-05 1.409780E-05 1.376792E-05 1.334501E-05 1.282974E-05 1.222769E-05 1.154469E-05
*    12 1.393349E-05 1.379708E-05 1.357151E-05 1.325471E-05 1.284745E-05 1.235277E-05 1.177366E-05 1.111763E-05
*    11 1.322233E-05 1.309336E-05 1.287981E-05 1.258004E-05 1.219378E-05 1.172463E-05 1.117661E-05 1.055492E-05
*    10 1.234630E-05 1.222616E-05 1.202766E-05 1.174842E-05 1.138966E-05 1.095306E-05 1.044343E-05 9.864016E-06
*     9 1.131043E-05 1.120070E-05 1.101984E-05 1.076573E-05 1.043902E-05 1.004169E-05 9.577058E-06 9.050248E-06
*     8 1.011875E-05 1.002138E-05 9.860499E-06 9.634454E-06 9.343773E-06 8.991042E-06 8.579095E-06 8.111635E-06
```

```
*     7 8.768968E-06 8.684338E-06 8.545007E-06 8.349889E-06 8.100176E-06 7.797165E-06 7.443588E-06 7.042870E-06
*     6 7.254398E-06 7.183459E-06 7.068451E-06 6.908355E-06 6.703606E-06 6.455625E-06 6.166968E-06 5.840204E-06
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     2 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     1 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*
*         r mesh  9   r mesh  10   r mesh  11   r mesh  12   r mesh  13   r mesh  14   r mesh  15   r mesh  16
*
*    30 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    29 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    28 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    27 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    26 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    25 5.478502E-06 5.086241E-06 4.666726E-06 4.225007E-06 3.762507E-06 3.288026E-06 2.793175E-06 0.000000E+00
*    24 6.599396E-06 6.116369E-06 5.599625E-06 5.051971E-06 4.477180E-06 3.881695E-06 3.242142E-06 0.000000E+00
*    23 7.592921E-06 7.027966E-06 6.422217E-06 5.778312E-06 5.106069E-06 4.404237E-06 3.652694E-06 0.000000E+00
*    22 8.464640E-06 7.827537E-06 7.143679E-06 6.420431E-06 5.666129E-06 4.874149E-06 4.025850E-06 0.000000E+00
*    21 9.222127E-06 8.524319E-06 7.775270E-06 6.984878E-06 6.156580E-06 5.283298E-06 4.351876E-06 0.000000E+00
*    20 9.866701E-06 9.116538E-06 8.312256E-06 7.462628E-06 6.569796E-06 5.630243E-06 4.630497E-06 0.000000E+00
*    19 1.039031E-05 9.596786E-06 8.747566E-06 7.850246E-06 6.906293E-06 5.912820E-06 4.857133E-06 0.000000E+00
*    18 1.078612E-05 9.961678E-06 9.079791E-06 8.144984E-06 7.161024E-06 6.127315E-06 5.029908E-06 0.000000E+00
*    17 1.105383E-05 1.020847E-05 9.302449E-06 8.342442E-06 7.332852E-06 6.271834E-06 5.146276E-06 0.000000E+00
*    16 1.118778E-05 1.033184E-05 9.414678E-06 8.442161E-06 7.418963E-06 6.344537E-06 5.204874E-06 0.000000E+00
*    15 1.118778E-05 1.033184E-05 9.414678E-06 8.442161E-06 7.418963E-06 6.344537E-06 5.204874E-06 0.000000E+00
*    14 1.105382E-05 1.020847E-05 9.302449E-06 8.342442E-06 7.332851E-06 6.271834E-06 5.146276E-06 0.000000E+00
*    13 1.078612E-05 9.961677E-06 9.079790E-06 8.144983E-06 7.161023E-06 6.127315E-06 5.029908E-06 0.000000E+00
*    12 1.039031E-05 9.596784E-06 8.747565E-06 7.850245E-06 6.906292E-06 5.912819E-06 4.857132E-06 0.000000E+00
*    11 9.866699E-06 9.116536E-06 8.312255E-06 7.462627E-06 6.569795E-06 5.630242E-06 4.630497E-06 0.000000E+00
*    10 9.222124E-06 8.524317E-06 7.775268E-06 6.984877E-06 6.156579E-06 5.283297E-06 4.351875E-06 0.000000E+00
*     9 8.464637E-06 7.827535E-06 7.143677E-06 6.420430E-06 5.666128E-06 4.874148E-06 4.025850E-06 0.000000E+00
*     8 7.592919E-06 7.027964E-06 6.422216E-06 5.778312E-06 5.106069E-06 4.404237E-06 3.652694E-06 0.000000E+00
*     7 6.599394E-06 6.116367E-06 5.599624E-06 5.051970E-06 4.477179E-06 3.881695E-06 3.242142E-06 0.000000E+00
*     6 5.478500E-06 5.086240E-06 4.666725E-06 4.225006E-06 3.762507E-06 3.288027E-06 2.793175E-06 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     2 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     1 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*
*         r mesh  17   r mesh  18   r mesh  19   r mesh  20   r mesh  21   r mesh  22   r mesh  23   r mesh  24
*
*    30 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    29 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    28 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    27 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    26 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    25 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    24 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    23 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    22 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    21 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    20 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    19 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    18 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    17 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    16 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    15 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    14 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    13 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    12 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    11 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*    10 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     9 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     8 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     7 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     2 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     1 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*
*         r mesh  25
*
*    30 0.000000E+00
*    29 0.000000E+00
*    28 0.000000E+00
*    27 0.000000E+00
*    26 0.000000E+00
*    25 0.000000E+00
*    24 0.000000E+00
*    23 0.000000E+00
*    22 0.000000E+00
*    21 0.000000E+00
*    20 0.000000E+00
*    19 0.000000E+00
*    18 0.000000E+00
*    17 0.000000E+00
*    16 0.000000E+00
*    15 0.000000E+00
*    14 0.000000E+00
*    13 0.000000E+00
*    12 0.000000E+00
*    11 0.000000E+00
*    10 0.000000E+00
*     9 0.000000E+00
*     8 0.000000E+00
*     7 0.000000E+00
*     6 0.000000E+00
*     5 0.000000E+00
*     4 0.000000E+00
*     3 0.000000E+00
*     2 0.000000E+00
*     1 0.000000E+00
*
*
*                                    ...interface file rtflux written..
```

```
*
*
*                                   ...interface file snconx written..

twodant iteration time, mins  6.9626E-02
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                    edit run on 11/15/94 with solver version 10-13-94*product release 2.5      machine d
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                                ...edit output...
*
*****************************************************************************************************************************************
*
*
*                                             ...block vi - edit specification data...
*
*
************************
*key start edit output *
************************


*************************************************************************************************************
*
*
* cross section balancing (balxs.ne.0)
*       or
* transport correction (trccr>diag, cesaro, or bhs)
* will NOT be reflected in edits
*
*
*************************************************************************************************************


*                                        ...input control integers...
*
*                 0  pted      0/1     no/yes - point edits desired
*                 1  zned      0/1     no/yes - zone edits desired
*
*                 0  ajed      0/1     direct/adjoint edit (use rtflux/atflux file)
*
*                 0  igrped    0/1/2/3 print totals only/print broad groups only/same as 1/print all groups and totals
*                 0  byvolp    0/1     no/yes - multiply point reaction rates by mesh volumes
*                 0  rzflux    0/1     no/yes - write the rzflux file (zone average flux file)
*                 0  rzmflx    0/1     no/yes - write the rzmflx file (zone average flux moments file)
*
*
*                                        ...floating parameters...
*
*          0.000000E+00  power     0/p     no/normalize all results, including flux files, to p megawatts
*          2.100000E+02  mevper    mev per fission (default: 210 mev)
*
*
*
*                                    ...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 broad groups
*
*
*
*                                    ...space related edit information...
*
*                                      0  number of points to edit
*                                      6  number of zones to edit
*                                      0  0/1  no/yes  density factors were input
*                                      1  0/1/2/3  no/solver/edit/both  zones mass edit
*                                           (requires atomic weights to be present)
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
```

```
...edit output...

...edit specification data(continued)...

...description of cross section edits...

isotope no.     material no.     constituent no.
-------- ---    --------- ---    ---------- ---

-none-          resdnt           -none-

reaction rates will be formed for each
of the above
using the cross section types shown below

                position
type            no.
-------         ---
chi             1
nusigf          2
total           3
abs             4
edit1           5

*key start    materials    *

reaction rates
from
materials

...zone edit for the sum of the neutron groups ...

... resident macroscopic ...

zone    volume      chi          nusigf       total        abs          edit1

  1 4.2412E+04  0.00000E+00  0.00000E+00  8.77851E-01  2.26081E-03  0.00000E+00
  2 5.3014E+04  0.00000E+00  0.00000E+00  3.59870E-01  9.16557E-04  0.00000E+00
  3 1.2723E+05  0.00000E+00  9.80630E-01  1.22668E+01  5.39537E-01  3.18645E-01
  4 1.5904E+05  0.00000E+00  0.00000E+00  2.68060E+00  6.89544E-03  0.00000E+00
  5 4.2412E+04  0.00000E+00  0.00000E+00  8.77852E-01  2.26081E-03  0.00000E+00
  6 5.3014E+04  0.00000E+00  0.00000E+00  3.59870E-01  9.16558E-04  0.00000E+00
  ----------    ----------   ----------   ----------   ----------   ----------
sum 4.7713E+05  0.00000E+00  9.80630E-01  1.74228E+01  5.52787E-01  3.18645E-01
```

```
*
*
*
*****************************
*key start run highlights    *
*****************************
*                                        run highlights
*
*                                  ***************************************
*                                  *                                     *
*                                  *    all modules are tentatively go.  *
*                                  *     interface file geodst written.  *
*                                  *       cross sections from cards.    *
*                                  *     interface mixing files written. *
*                                  *      interface file asgmat written  *
*                                  *    xs files macxs,srcedt written.   *
*                                  * *left boundary condition overridden* *
*                                  *      interface file solinp written. *
*                                  *      interface file editit written. *
*                                  *         start solver execution.     *
*                                  *     all convergence criteria met.   *
*                                  *   fixsrc written with fissions      *
*                                  *      interface file rtflux written. *
*                                  *      interface file srcnm written.  *
*                                  *           start edit execution.     *
*                                  *              edits completed.       *
*                                  *                                     *
*                                  ***************************************
*
*
*                                        storage and timing history
*                                  ***********************************************************
*                                  *          scm      scm      lcm      lcm      cpu     sysio *
*                                  *  module  words    limit    words    limit    seconds seconds *
*                                  *  ------  -------  -------  -------  -------  ------- ------- *
*                                  *       0        0        0        0        0     4.4     0.4 *
*                                  *     100     8539    40000        0        0     3.7     0.2 *
*                                  *     101        0        0        0        0     0.7     0.0 *
*                                  *     102     1043    40000       10   140000     0.0     0.0 *
*                                  *     103       50    40000        0        0     0.0     0.0 *
*                                  *     104     2065    40000        0        0     0.0     0.1 *
*                                  *     105        0        0        0        0     0.0     0.0 *
*                                  *     106        0        0        0        0     0.0     0.0 *
*                                  *     107      191    40000       83   140000     0.0     0.0 *
*                                  *     108      592    40000        7   140000     0.0     0.0 *
*                                  *     109     4429    40000        0        0     0.0     0.0 *
*                                  *     112        0        0        0        0     0.0     0.0 *
*                                  *     200    10997    40000    12689   140000     0.4     0.1 *
*                                  *     201        0        0        0        0     0.0     0.0 *
*                                  *     202        0        0        0        0     0.0     0.0 *
*                                  *     203      927    40000        0        0     0.0     0.0 *
*                                  *     204        0        0        0        0     0.0     0.0 *
*                                  *     205        0        0        0        0     0.0     0.0 *
*                                  *     206        0        0        0        0     0.2     0.0 *
*                                  *     207        0        0        0        0     0.1     0.0 *
*                                  *     208        0        0        0        0     0.0     0.0 *
*                                  *     210        0        0        0        0     0.1     0.0 *
*                                  *     211        0        0        0        0     0.0     0.0 *
*                                  *     300        0        0        0        0     0.2     0.1 *
*                                  *     301     7481    40000       61   140000     0.1     0.0 *
*                                  *     302        0        0        0        0     0.0     0.0 *
*                                  *     400        0        0        0        0     0.0     0.0 *
*                                  ***********************************************************
*
*
*                                        ...execution terminated...
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
```

Sample Problem 2: Coupled S_n Monte Carlo Calculation

Sample Problem 2 is a coupled Monte Carlo/S_n calculation for a two-dimensional cylindrical shielding duct. Two energy-group cross sections are used and the scattering is assumed to be linearly anisotropic.

The cylindrical shielding duct model consists of a 5 cm radius void, a 2 cm thick iron wall, and a 18 cm thick iron-water shield (80 v/o iron and 20 v/o water). The length of the duct is 50 cm. An isotropic neutron flux in group 1 is assumed to be present at the bottom of the duct. Because of the neutron streaming in the duct, this problem is difficult to calculate using standard S_n. And, because of multiple scattering in the shield region, it is expensive to calculate using just Monte Carlo. Thus, by using Monte Carlo in the duct region, and S_n in the shield region, we can achieve a more efficient solution than can either technique by itself.

Sample Problem 2: Output Description

The Monte Carlo input variables as defaulted by the code or as overridden by the user, are echoed in the Monte Carlo Input Parameters table (page 3-110). The Monte Carlo Setup Information table details the additional memory requirements for the MC/S_n method, the group-dependent Monte Carlo/S_n boundaries in terms of fine mesh cells, and the resulting fraction of the fixed source located in the Monte Carlo region. Additional LCM is needed to read in the cross sections for the Monte Carlo, and to provide storage for the interface fluxes (residuals) along the MC/S_n boundary. If insufficient LCM is present, the residuals will be stored on disk by energy group.

For this sample problem, the duct (fine mesh cells 1 through 5) has been designated as the Monte Carlo region by use of the input variable MCIBND. Since MCJBND was not entered, the Monte Carlo region extends the entire problem length. This would place the Monte Carlo boundary directly along the duct/wall interface. However, the angular flux at such material interfaces is usually a strong function of angle and would require a large S_n order to resolve. To allow the use of a lower S_n order, the code automatically moves the Monte Carlo/S_n interface MCBLT mean free paths into the S_n region. Thus, since the mean free path is group dependent, the actual Monte Carlo/S_n interface location will be a function of energy group, and differ from that specified by MCIBND/MCJBND, unless MCBLT is set to zero by the user. In this problem, the Monte Carlo/S_n method is only used in energy group 1, so no boundaries are listed for energy group 2.

The Monte Carlo region may consist of any or all energy groups in a problem. It is generally best to specify the Monte Carlo/S_n option in only the source and/or higher energy groups, since S_n is more efficient in the lower energy groups where scattering predominates. Note that by assigning all energy and spatial regions to the Monte Carlo region, it is possible to run a multigroup Monte Carlo calculation with TWODANT.

The Monte Carlo calculations are performed using the macroscopic cross sections from the interface file MACRXS, so that no additional cross-section input is required. In

order to represent angular scattering for the group-to-group transfer matrices, TWODANT generates 32 equiprobable bins for each scattering matrix such that the moments of the angular scattering cross sections are conserved. This generation is performed via a maximum entropy method (see Chapter 13 for more details). If problems are encountered in generating the bins from the cross-section moments, they are listed in the MC Angular Bin Generation table (page 3-110). Generally it is best to use as many moments as possible when generating the equiprobable scattering bins, so ISCT should be set to the maximum number of moments available from the cross-section tables.

After generation of the scattering bins, the fixed source located in the Monte Carlo region (i.e., the isotropic surface flux along the bottom of the duct) is sampled using MCNTR batches of (approximately) MCNHIS histories each. The resulting Figure of Merit (FOM) for the self scatter (collision) rate over the Monte Carlo region for each energy group is printed in the Self Scatter FOM, Source Calculation table on page 3-111. The FOM (defined as one over the variance times the time) should approach a constant value for a well sampled problem. For this problem, we see a large change in the FOM between 140280 and 150300 histories, and another one between 210420 and 220440 histories. This indicates that MCNTR and/or MCNHIS should be increased for a more accurate sampling of the Monte Carlo region.

Additional indications of how well the fixed source inside the Monte Carlo is being sampled are given in the MC Coarse Mesh Tracklengths table, the MC Coarse Mesh Leakages table, and the Tracks Entering (Coarse Mesh) table (page 3-111). The MC Coarse Mesh Tracklengths table gives the total tracklength per coarse mesh due to the fixed source, along with an estimate of the relative error. Note that since the third coarse mesh is entirely in the S_n region, particles are not tracked through it. The MC Coarse Mesh Leakages table is the leakage (particles/sec) from each coarse mesh cell along the outer edge of the Monte Carlo region, along with an estimated relative error. Here, the right leakage represents the leakage from the Monte Carlo region into the S_n (shield) region due to the surface flux at the duct bottom. The Tracks Entering (Coarse Mesh) table gives the number of tracks that enter or cross a coarse mesh cell. Note that a particle may scatter and re-enter a cell more than once, so the number of tracks entering a cell may be larger than the number of source histories.

The MC Source Calculation Particle Balance table (page 3-112) entries are as defined for the S_n balance table entries, except that the fission source and in scatter entries are broken out as to amounts that appear in the S_n region and the Monte Carlo region. The net leakage entry reflects the leakage from the Monte Carlo region. For problems with splitting/Russian roulette, the particle balance may not approximate zero as closely, since these techniques create/destroy weight. However, the weight balance entry also shown in this table should still approximately equal machine precision, since it accounts for effects due to variance reduction techniques. If there is splitting/Russian roulette in the problem, due to cell (MCCMIMP) or energy group (MCEGIMP) importances, or due to weight splitting from fission events, the number of particles (and weight) created/destroyed are printed here.

Particles created by splitting are placed into a particle bank for subsequent tracking. The number of particles the bank can hold is initially sized at 640. If the bank fills up, more space is allocated to it, within the limits of MAXLCM. The maximum size the bank actually reaches during a Monte Carlo calculation is printed out here. Since there is no splitting in this problem, the maximum bank number is zero.

Once the Monte Carlo source calculation has been completed, a normal S_n calculation is performed for the entire problem geometry, except that the Monte Carlo calculated boundary flux at the shield/wall interface is used as an internal boundary condition. In this problem, since there is no upscatter or fission, the S_n calculation consists of one outer iteration (page 3-113). Since there is no need to determine the fluxes in the second energy group until the boundary flux at the Monte Carlo/S_n interface has been converged, the S_n calculation for group 2 is skipped. Also note that the S_n particle balance is not determined until after the boundary flux has been converged.

Next, the angular boundary fluxes from the S_n region into the MC region at the Monte Carlo/S_n interface are sampled to determine the response of the Monte Carlo region due to the coupling with the S_n region. This process of coupling an S_n calculation and a Monte Carlo calculation is referred to as a "super-outer" iteration. The Monte Carlo/S_n method in TWODANT is "fully coupled" in the sense that it iterates between the S_n and Monte Carlo regions until the interface fluxes are converged. Information about the coupling, or link, calculations is contained in the Sitno (Super-outer Iteration) Link Monitor tables (page 3-114 through page 3-119).

The convergence criteria for the S_n/Monte Carlo iterations is based upon the errors in the group-dependent scalar boundary fluxes at the spatial S_n/MC interface (i.e., the shield-wall interface), and the group-dependent scalar scattering/fission sources, in this case the downscatter in the wall from the group 1 Monte Carlo to the group 2 S_n region. Convergence in super-outers is considered to have been achieved when the relative error in the Monte Carlo to S_n boundary flux and the Monte Carlo to S_n scattering/fission sources is less than EPSI when compared from one super-outer iteration to the next. These errors are printed out in the Sitno Link Monitor tables as the "iteration error." In addition, the energy group location of the maximum errors as well as a cell index are printed out.

In order to accelerate convergence of the super-outer iterations, the interface fluxes and sources (if any) from the S_n region into the Monte Carlo region are decomposed into orthonormal "basis vectors" (shapes), where each new S_n calculation defines an additional basis vector. The basis vectors are what are actually sampled to determine an associated "response vector" for the Monte Carlo region, where each Monte Carlo coupling calculation determines one response vector. How well the existing basis vectors represent the current S_n to Monte Carlo interface fluxes/sources is represented by the "fit errors" in the Sitno Link Monitor tables. The "current fit coefficients" are the coefficients used to fit the basis vectors to the interface fluxes/sources. After a certain number of super-outer iterations (five in this case), the existing basis vectors are sufficient to describe the interface fluxes and sources from the S_n region into the Monte Carlo region to within EPSI (page 3-119). Then, the existing basis vectors may be projected (see Chapter 13 for more details) to determine the converged fit coefficients without any further Monte Carlo coupling calculations.

After projection, one additional super-outer calculation is done, with the S_n calculation using the projected interface fluxes/sources. Now, a complete S_n calculation (all energy groups and particle balance) is performed. We see on page 3-119 that the resulting iteration errors in the Monte Carlo to S_n boundary flux and the Monte Carlo to S_n scattering/

fission sources are within our error criteria EPSI (.001), while the fit error of the basis vectors to the S_n to Monte Carlo interface fluxes/sources is reasonably good.

The number of histories used in each Monte Carlo link calculation is determined automatically by the code, and is based upon the number of histories specified for the source calculation and the relative magnitude of the current fit coefficients. Thus, if we examine the number of histories used in the Sitno Link Monitor tables on page 3-114 through page 3-119, we see that as the magnitude of the current fit coefficients decline, so does the number of histories used. However, at least one particle per trial is started for each phase space cell along the Monte Carlo/S_n interface (see Chapter 13 for more details). This defines a lower limit on the number of histories used, which is 6000 for this sample problem, and is the number of histories used for the third and fourth link calculations. For problems with a larger Monte Carlo/S_n interface, and/or a larger value of MCNTR, this lower limit could be considerably higher.

After the convergence of the super-outer iteration, the Monte Carlo Region Particle Balance Table (page 3-119) is recalculated. However, the entries in this table are a bit different from those on page 3-112. Here, SOURCE is again the fixed source inside the Monte Carlo region, while S_n SOURCE refers to in scatter/fission sources from the S_n region into the Monte Carlo region. FISSION SRC and IN SCATTER refer to in scatter and fission sources from the Monte Carlo region to the Monte Carlo region. The NET LEAKAGE is the leakage (particles/sec) from the Monte Carlo region minus the leakage from the S_n region into the Monte Carlo region. Thus, this quantity is lower here than that reported on page 3-112.

The MC Coarse Mesh Leakages table (page 3-119) gives the leakage from the Monte Carlo region, not the net leakage. Since a particle may scatter across the Monte Carlo/S_n interface multiple times, this leakage can be greater than the source, as is the case here. It is interesting to compare this table with that on page 3-111 to see the effects on the leakage of coupling with the S_n region.

The Tracks Entering (Coarse Mesh, Rm Calc) table (page 3-119) is the number of tracks that enter or cross a coarse mesh for just the Monte Carlo coupling calculations. The relative fractions in coarse meshes one and two differ in this table from that on page 3-111 since the histories for the link calculations originate at the S_n/Monte Carlo interface, i.e., the shield-wall boundary, while those for the fixed source calculation originate at the duct bottom.

The Balance Table (page 3-120) is as defined for regular S_n calculations, and incorporates the balance for both the S_n and Monte Carlo regions. This is also true for the Coarse Mesh Balances table page 3-121 output specified by the BALP option. This table can be used to determine the leakages out the top of the duct, as well as the shield and wall regions. Note that for this problem, the top leakages from the duct and wall regions are available from the MC Coarse Mesh Leakages table on page 3-119 as well. However, if the coarse mesh cell defining the wall was split between the S_n and Monte Carlo regions, the MC Coarse Mesh Leakages table would only specify the leakage from the Monte Carlo portion of the cell, while the Coarse Mesh Balances table would combine both the S_n and Monte Carlo leakages.

Although no edits were specified for this problem, they are still possible with the S_n/Monte Carlo option. Flux estimates for the Monte Carlo region are provided through a tracklength estimator, and these fluxes are used in the EDIT module. Since flux estimates are only available on the coarse grid, however, an average value is used for all fine mesh cells in a coarse mesh cell in the Monte Carlo region. If one desires more flux detail inside the Monte Carlo region, one must refine the coarse mesh structure, with a corresponding increase in the computational time required for the Monte Carlo. If a coarse mesh cell is split between the Monte Carlo and S_n regions, Monte Carlo calculated fluxes are used only in the Monte Carlo portion of the coarse mesh cell.

Sample Problem 2: Output Listing

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*            generalized input module run on 12/ 6/94 with solver version 10-13-94beta---- release 2.6b     machine bowline
*
*****************************************************************************************************************************************
*                  ***************************************************************************************
*                  *                                                                                     *
*                  *                    ...listing of cards in the input stream...                       *
*                  *                                                                                     *
*                  *    1.     1          0                                                              *
*                  *    2.   sample sn/mc problem for twodant user's guide                               *
*                  *    3. /*************************** B L O C K  I ****************************        *
*                  *    4. igeom=r-z  ngroup=2  isn=4                                                    *
*                  *    5. im=3  jm=1  it=25  jt=40                                                      *
*                  *    6. nzone=2  mt=2  niso=2                                                         *
*                  *    7.   t                                                                           *
*                  *    8. /*************************** B L O C K  II ***************************        *
*                  *    9. xints=5 2 18  yints=40                                                        *
*                  *   10. xmesh=0. 5. 7. 25.                                                            *
*                  *   11. ymesh=0. 50.                                                                  *
*                  *   12. zones=0 1 2                                                                   *
*                  *   13.   t                                                                           *
*                  *   14. /*************************** B L O C K  III **************************        *
*                  *   15. lib=cdnimp                                                                    *
*                  *   16. maxord=1  ihm=6  iht=3  ihs=4  ititl=1                                        *
*                  *   17. names=wall shield                                                             *
*                  *   18.   t                                                                           *
*                  *   19. wall, p0                                                                      *
*                  *   20.  0.01        0.0         0.78        0.75        0.0         0.0              *
*                  *   21.  0.22        0.0         1.14        0.92        0.02        0.0              *
*                  *   22. wall, p1                                                                      *
*                  *   23.  0.0         0.0         0.0         0.03        0.0         0.0              *
*                  *   24.  0.0         0.0         0.0         0.01        0.0         0.0              *
*                  *   25. shield, p0                                                                    *
*                  *   26.  0.01        0.0         0.84        0.81        0.0         0.0              *
*                  *   27.  0.18        0.0         1.32        1.14        0.02        0.0              *
*                  *   28. shield, p1                                                                    *
*                  *   29.  0.0         0.0         0.0         0.15        0.0         0.0              *
*                  *   30.  0.0         0.0         0.0         0.26        0.0         0.0              *
*                  *   31. /*************************** B L O C K  IV ***************************        *
*                  *   32. matls=isos                                                                    *
*                  *   33. assign=matls                                                                  *
*                  *   34.   t                                                                           *
*                  *   35. /*************************** B L O C K  V ****************************        *
*                  *   36. ievt=0  epsi=.001  isct=1                                                     *
*                  *   37. ibl=1  ibr=0  ibb=0  ibt=0                                                    *
*                  *   38. xsectp=2  norm=1.0  balp=1                                                    *
*                  *   39. sibott=1 0; 4y1; 2r0; 19y1                                                    *
*                  *   40. source=2r0;2y1                                                                *
*                  *   41. mcopt=1 mcreg=1 0 mcibnd=1 5                                                  *
*                  *   42.   t                                                                           *
*                  *                                                                                     *
*                  ***************************************************************************************
```

```
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                                    case title
*
*********************************************************************************************************************************
************************
*key start case input  *
************************
*                                  1  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 sn/mc problem for twodant user's guide                         *
*                          *                                                                         *
*                          ***************************************************************************
************************
*key end   block i read*
************************
*********************************************************************************************************************************

*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                          ...block i - controls and dimensions...
*
*********************************************************************************************************************************
*
*
*                                               ...dimensions (array name = dimens)...
*
*                           7  igeom   6/7/9/11 x-y/r-z/triangles/r-theta
*                           2  ngroup  number of energy groups
*                           4  isn     angular quadrature order
*                           2  niso    number of input isotopes (from isotxs, grupxs, or cards)
*                           2  mt      number of permanent materials
*                           2  nzone   number of zones
*                           3  im      number of coarse mesh x intervals
*                          25  it      number of fine mesh intervals
*                           1  jm      number of coarse mesh y intervals
*                          40  jt      number of fine mesh y intervals
*
*                                               ...storage...
*
*                                              maxdcm=  140000
*                                              maxscm=   40000
*
*********************************************************************************************************************************
******************************
*key end   block ii read-geom*
******************************
*
*                         7  igeom    1/2/3/6/7/8/9/11/14
******************************
*key end   block iii read-xs *
******************************
```

```
...block iii - cross section library...

...library source...
lib=cdnlnp

...card library parameters (array name = cards)...

1  maxord  maximum legendre order to be found in input cross sections
6  ihm     last position in cross section table
3  iht     position of total cross section
4  ihs     position of self scatter cross section
0  ifido   -1/0/1/2 - dlf(4e18.0)/dlf/fixed fido/free fido library
1  ititl   0/1 - no/yes there is a title card before each table
0  i2lp1   0/1 - no/yes library higher order scattering contains 21+1 factor
0  savbxs  0/1 - no/yes save binary xslib file (filename=bxslib)
1  kwikrd  0/1 - full fido read/quick fido read (default=quick)

...energy structure...

group    chi          vel          lower bound   upper bound   group    chi          vel          lower bound   upper bound
-----  ---------  ---------  ---------  ---------  -----  ---------  ---------  ---------  ---------
    1  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00      2  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00

last neutron group(lng) is number   2


0  balxs   -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance

...edit position names...

                                    card
position  edname  position
--------  -----  --------
       1  chi
       2  nusigf        2
       3  total         3
       4  abs           1

*******************************
*key start card libe read     *
*******************************

...header cards from the card library...

isotope  isotope
number   name   order      header card
------  ------  -----  ---------------------
    1.  wall    p0     -  wall, p0
                p1     -  wall, p1
    2.  shield  p0     -  shield, p0
                p1     -  shield, p1

*******************************
*key end   card libe read     *
*******************************
*******************************
*key end   block iv read-mats*
*******************************
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                        ... mixing instructions ...
*****************************************************************************************************************************************
*
*          mix      comp       density    comp       density etc.
*          ---      ----       -------    ----       -------
*
*     matls
*     -----
*
*       1. wall     wall     1.00000E+00,
*       2. shield   shield   1.00000E+00,
************************
*key start mix card xs *
************************
************************
*key end   mix card xs *
************************
*****************************************************************************************************************************************
*****************************
*key end   block v read-solvr*
*****************************
************************
*key end   input module*
************************
*
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
                    this twodant problem run on 12/ 6/94 with solver version 11-23-94beta---- release 2.6b      machine bowline
*  sample sn/mc problem for twodant user's guide
*
*****************************************************************************************************************************************
*
*                                        ...block v -- solver input...
*
*****************************************************************************************************************************************
*          raw     as
*         input  defaulted
*
*                                        ...required input(array name = solin)...
*
*          0        0       ievt   0/1/2/3/4 - type of calculation
*                                      0 inhomogeneous source
*                                      1 k-effective
*                                      2 alpha or time absorption search
*                                      3 concentration search
*                                      4 delta(i.e. dimension) search
*          1        1       isct   legendre order of scattering
*          0        0       ith    0/1 - direct/adjoint - mode of calculation (default=direct)
*          1        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
*          0        0       ibt    0/1/2/3 - top boundary condition
*                                     vacuum/reflective/periodic/white
*          0        0       ibb    0/1/2/3 - bottom boundary condition
*                                     vacuum/reflective/periodic/white
*
*
*                                        ...convergence controls(array name = iter)...
*
*      1.000E-03 1.000E-03  epsi   inner iteration convergence criterion (default=0.0001)
*          0        1       iitl   maximum number of inner iterations per group until fission source is near
*                                  convergence, i.e. lambda is near convergence. (default=1)
*          0       30       iitm   maximum number of inner iterations per group when close to fission source convergence
*                                  (default calculated)
*          0       20       oitm   maximum number of outer iterations (default=20)
*          0        0       itlim  iteration time limit (seconds)
*
*
*                                        ...miscellaneous parameters...
*
*                   0       nosigf set fissions zero when source calculation (0/1 no/yes)
*                   0       asleft write in-going angular boundary flux to file arbflux (0/i no/yes)
*                   0       asrite write out-going angular boundary flux to file arbflux (0/i no/yes)
*                   0       astop  write up-going angular boundary flux to file arbflux (0/j no/yes)
*                   0       astott write down-going angular boundary flux to file arbflux (0/j no/yes)
*                   0       iufcs  fcsrc option (0/1/2 no/analytic/ray tracing)
*
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                   ...block v -- solver input (continued)...
*
*****************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*
*                                        ...miscellaneous parameters(array name = misc)...
*
*      0.000E+00 0.000E+00  bhgt   buckling height
*      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     10000     fcnray number of rays/trial for fcsrc rt option
*           0        25     fcntr  number of trials for fcsrc rt option
*           0         1     iquad  -3/-2/1/2/3 - source of quadrature constants (default=1)
*                              -3 sncon file
*                              -2 hybrid product set (triangular arrangement)
*                               1 old twotran built-in set
*                               2 product set (rectangular arrangement)
*                               3 card input
*
*
*                                           ...output controls(array name = solout)...
*
*                       0  fluxp  0/1/2   none/isotropic/all moments - flux print
*                       2  xsectp 0/1/2   none/principal/all - macroscopic cross section print
*                       0  fissrp 0/1     no/yes - print final fission source rate
*                       0  sourcp 0/1/2/3 no/as read/normalized/both - print inhomogeneous source
*                       0  angp   0/1     no/yes - print angular fluxes
*                       0  raflux 0/1     no/yes - write angular fluxes to file raflxm or aaflxm(if ith=1)
*                       0  rmflux 0/1     no/yes - write flux moments to file rmflux
*                       1  balp   0/1     no/yes - print coarse mesh balances
*
*
*                                           ...parameters inferred from input arrays...
*
*                                  0  inchi  0/1/2 none/one chi/zonewise chi
*                                  0  isdenx 0/1/n - none/x density vector/full matrix
*                                  0  isdeny 0/1 no/yes - use y density vector
*                                  1  iqan   source anisotropy
*                                  3  isorse number of source moments input
*                                  0  isorsx number of sourcx moments input
*                                  0  isorsy number of sourcy moments input
*                                  0  isorsf number of sourcf moments input
*                                  0  iql    -1/0/1/2 isotropic/none/all angles/vectors -left boundary source
*                                  0  iqr    -1/0/1/2 isotropic/none/all angles/vectors -right boundary source
*                                  0  iqt    -1/0/1/2 isotropic/none/all angles/vectors -top boundary source
*                                 -1  iqb    -1/0/1/2 isotropic/none/all angles/vectors -bottom boundary source
*
*
*
*****************************************************************************************************************************
*****************************************************************************************************************************


*****************************************************************************************************************************
*****************************************************************************************************************************
*
*
*                                        ...parameters from block i...
*
*                                 7  igeom  6/7/11  x-y/r-z/r-theta
*                                 2  ngroup number of energy groups
*                                 4  isn    angular quadrature order
*                                 2  mt     number of permanent materials
*                                 2  nzone  number of zones
*                                 3  im     number of coarse mesh x intervals
*                                 1  jm     number of coarse mesh y intervals
*                                25  it     number of fine mesh x intervals
*                                40  jt     number of fine mesh y intervals
*
*
*
*****************************************************************************************************************************
*****************************************************************************************************************************
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*
*                                        ...material assignments to zones...
*
*
*******************************
*key start matls to zones     *
*******************************
*
*                                                zone     material
*                                              no. name   no. name
*
*                                               1 zone1    1 wall
*                                               2 zone2    2 shield
************************
*key start storage map *
************************
*
*  scm storage summary...
*
*     total scm required for this problem       16000
*     maximum scm available (maxscm= )          40000
*     scm required for transport                15961
*     scm required for diffusion                15978
*
*
*  lcm storage summary...
*
*     lcm selected for this problem                30819
*     maximum lcm specified (maxlcm= )            140000
*          Storage required for all quantities  in core is   30819
*
*          source moments   are in core.  If O.K. to put in core, set MAXLCM .ge.    30819
*          diffusion params are in core.  If O.K. to put in core, set MAXLCM .ge.    28819
*          flux moments     are in core.  If O.K. to put in core, set MAXLCM .ge.    25494
*          scalar source    are in core.  If O.K. to put in core, set MAXLCM .ge.    23494
*          diffusion fluxes are in core.  If O.K. to put in core, set MAXLCM .ge.    22185
*          scalar tr fluxes are in core.  If O.K. to put in core, set MAXLCM .ge.    21119
*          Storage required for all quantities  on disk is   19729
*


      30819 words lcm required lcmadd
*
*  zero flux
*
************************
*key start sn constants*
************************
*
*  s 4 constants for groups  1 to  2
*                 mu             eta            weight
*
*     1      0.30163878E+00  0.90444905E+00  0.83333332E-01
*     2      0.90444905E+00  0.30163878E+00  0.83333332E-01
*     3      0.30163878E+00  0.30163878E+00  0.83333332E-01
*****************************************************zones map*********************************************************
************************
*key start zones map   *
************************
      z                                 zones by coarse mesh, origin at lower left.
   rcw  m                               m is number of fine intervals per coarse mesh.
   50.0000  rvvvvvvvvvvvvvvvv
            r       *       *       v
     1  40  r   0   *   1   *   2   v
            r       *       *       v
    0.0000  rvvvvvvvvvvvvvvvv
      r      0.      5.      7.     25.
          0000    0000    0000    0000
      m          5       2      18
   column        1       2       3
```

```
...cross section related data from file macxs  000000100193 version  1 ...

 1 wall      2 shield

                                    ...cross sections for legendre orders up to p1...

*******************************
*key start mac cross sections*
*******************************

                                                  g r o u p   1

                                    ...principal cross sections...

                        zone          chi         nu*fission      total       absorption
                      no.  name
                       1  zone1     0.0000E+00    0.0000E+00    7.8000E-01    1.0000E-02
                       2  zone2     0.0000E+00    0.0000E+00    8.4000E-01    1.0000E-02

                                    ...scattering matrices...
                                       (21+1 not included)

   zone   order  first grp    cross sections
     1      0       1          7.5000E-01
     1      1       1          3.0000E-02
     2      0       1          8.1000E-01
     2      1       1          1.5000E-01

                                                  g r o u p   2

                                    ...principal cross sections...

                        zone          chi         nu*fission      total       absorption
                      no.  name
                       1  zone1     0.0000E+00    0.0000E+00    1.1400E+00    2.2000E-01
                       2  zone2     0.0000E+00    0.0000E+00    1.3200E+00    1.8000E-01

                                    ...scattering matrices...
                                       (21+1 not included)

   zone   order  first grp    cross sections
     1      0       2          9.2000E-01    2.0000E-02
     1      1       2          1.0000E-02
     2      0       2          1.1400E+00    2.0000E-02
     2      1       2          2.6000E-01
```

```
*key start nam. source*

...block_v — monte carlo input parameters...

          1  mcopt    0/1 no/yes - mc option
          8  mcits    max number of super-outers
          0  mciseed  initial random seed
      10000  mcnhis   number of hist/trial
         25  mcntr    number of trials
          1  mcilbnd  left mc boundary
          5  mcirbnd  right mc boundary
          0  mcjtbnd  bottom mc boundary
          0  mcjbnd   top mc boundary

  1.000E+00  mcblt     boundary layer thickness
  0.000E+00  mcwc      weight cutoff 1
  0.000E+00  mcwc      weight cutoff 2
  0.000E+00  mcsb      dist source biasing

                       ...parameters inferred from source arrays...

          0  imtsrc   0/1/2 dist/pt beam/pt source
          0  imbsla   0/1 no/yes - left beam source
          0  imbsra   0/1 no/yes - right beam source
          0  imbsta   0/1 no/yes - top beam source
          0  imbsba   0/1 no/yes - bottom beam source

...monte carlo setup information...

      30909 words lcm required mcxspt

 mc storage summary...

     lcm selected for this problem              68478
     maximum lcm specified (maxlcm= )          140000

     needed for all in lcm (no disk)            68478
     with residuals to disk                     52446

     residuals on disk?                         no

      68478 words lcm required mcxs

                                  ****************************************
                                  *  monte carlo fine mesh boundaries    *
                                  *                                      *
                                  * group   ilbl   irbl   jbbl   jtbl   *
                                  *   1       1      7      1     40    *
                                  *                                      *
                                  ****************************************

 the amount of the fixed source in the mc region is  1.000E+00

                       mc angular bin generation

 nbins =   32

 processing zone   1...

 processing zone   2...

 cpu time required =  6.667E-02 sec
```

```
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                        self scatter fom, source calculation
*
*********************************************************************************************************************************
*
*                group  1                                        group  2
*
*       nhis       mean     error     fom
*      10020   4.0237E+00 0.0000       0.0       10020  0.0000E+00 0.0000      0.0
*      20040   4.0105E+00 0.0044    1531.2       20040  0.0000E+00 0.0000      0.0
*      30060   4.0147E+00 0.0025    3072.3       30060  0.0000E+00 0.0000      0.0
*      40080   4.0209E+00 0.0022    3139.2       40080  0.0000E+00 0.0000      0.0
*      50100   4.0231E+00 0.0018    3805.4       50100  0.0000E+00 0.0000      0.0
*      60120   4.0300E+00 0.0024    1716.0       60120  0.0000E+00 0.0000      0.0
*      70140   4.0338E+00 0.0024    1447.9       70140  0.0000E+00 0.0000      0.0
*      80160   4.0312E+00 0.0021    1651.4       80160  0.0000E+00 0.0000      0.0
*      90180   4.0307E+00 0.0019    1875.8       90180  0.0000E+00 0.0000      0.0
*     100200   4.0230E+00 0.0025     988.2      100200  0.0000E+00 0.0000      0.0
*     110220   4.0223E+00 0.0022    1096.6      110220  0.0000E+00 0.0000      0.0
*     120240   4.0226E+00 0.0020    1209.9      120240  0.0000E+00 0.0000      0.0
*     130260   4.0189E+00 0.0021    1057.0      130260  0.0000E+00 0.0000      0.0
*     140280   4.0201E+00 0.0020    1131.1      140280  0.0000E+00 0.0000      0.0
*     150300   4.0244E+00 0.0021     904.2      150300  0.0000E+00 0.0000      0.0
*     160320   4.0230E+00 0.0020     939.5      160320  0.0000E+00 0.0000      0.0
*     170340   4.0204E+00 0.0020     899.9      170340  0.0000E+00 0.0000      0.0
*     180360   4.0225E+00 0.0019     892.7      180360  0.0000E+00 0.0000      0.0
*     190380   4.0198E+00 0.0020     833.0      190380  0.0000E+00 0.0000      0.0
*     200400   4.0203E+00 0.0019     878.5      200400  0.0000E+00 0.0000      0.0
*     210420   4.0201E+00 0.0018     923.3      210420  0.0000E+00 0.0000      0.0
*     220440   4.0235E+00 0.0019     775.5      220440  0.0000E+00 0.0000      0.0
*     230460   4.0215E+00 0.0019     756.0      230460  0.0000E+00 0.0000      0.0
*     240480   4.0234E+00 0.0019     739.4      240480  0.0000E+00 0.0000      0.0
*     250500   4.0221E+00 0.0018     750.2      250500  0.0000E+00 0.0000      0.0
*
*  absolute error (1 std dev) in integral ss is  7.2791E-03
*
*********************************************************************************************************************************
*********************************************************************************************************************************

*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                        mc coarse mesh tracklengths
*
*********************************************************************************************************************************
*
*         i       1   i       2   i       3
*     1  1.3473E+01  5.3662E+00  0.0000E+00
*          0.0017      0.0014      0.0000
*
*  xmax = 0.0017  i =   1  j =   1
*
*********************************************************************************************************************************
*********************************************************************************************************************************

*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                        mc coarse mesh leakages
*
*********************************************************************************************************************************
*
*                                          top   leakages
*
*         i       1   i       2   i       3
*        1.4307E-02  1.5236E-03  0.0000E+00
*          0.0207      0.0562      0.0000
*
*                                          bottom leakages
*
*         i       1   i       2   i       3
*        1.1052E-01  7.4257E-02  0.0000E+00
*          0.0055      0.0059      0.0000
*
*                                          right leakages
*
*         j       1
*        6.3810E-01
*          0.0011
*
*                                          left  leakages
*
*         j       1
*        0.0000E+00
*          0.0000
*
*********************************************************************************************************************************
*********************************************************************************************************************************

*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                                        tracks entering (coarse mesh)
*
*********************************************************************************************************************************
*
*         i       1   i       2   i       3
*     1       419747      387115           0
*
*********************************************************************************************************************************
*********************************************************************************************************************************
```

```
                                    mc source calculation particle balance

    gp          source       fiss src(mc)    fiss src(sn)   in scatter(mc)  in scatter(sn)

     1     1.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00
     2     0.0000E+00     0.0000E+00     0.0000E+00    -2.7756E-17     1.0766E-01
   tot     1.0000E+00     0.0000E+00     0.0000E+00    -2.7756E-17     1.0766E-01

    gp       self scatter     net leakage     absorption     out scatter   particle balance

     1     4.0221E+00     8.3871E-01     5.3634E-02     1.0766E-01     1.6365E-13
     2     0.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00     0.0000E+00
   tot     4.0221E+00     8.3871E-01     5.3634E-02     1.0766E-01     1.6365E-13


                                    ... source calculation information ...

 number of histories, source calculation      =    250500
 cpu time required =  4.070E+02 sec
 avg. no. of coll/hist =   4.39
 avg. source particle weight =    0.9980
 min source particle weight  =    0.9759
 max source particle weight  =    0.9998
 weight balance =  1.636E-13


 particles generated through weight splitting =         0
 weight gain =  0.000E+00
 weight loss =  0.000E+00

 the maximum bank number was       0
```

```
*******************************************************************************************************************************
*******************************************************************************************************************************
*
*                                    ...iteration controls and criteria...
*
*******************************************************************************************************************************
*
*
*                                          ***iteration criteria***
*
*                                                 transport inners
*
*          criterion    quantity to test                                  value     action taken if value exceeded
*          ---------    ----------------                                  -----     ------------------------------
*             iitl  - inner iteration count until near lambda             30        terminates inners
*                     (i.e. fission source) convergence
*             iitm  - inner iteration count when near lambda              30        terminates inners
*                     (i.e. fission source) convergence
*             epsi  - fractional ptwise flux change per inner          1.00E-03     does another inner
*
*                                              diffusion sub-outers
*
*          criterion    quantity to test                                  value     action taken if value exceeded
*          ---------    ----------------                                  -----     ------------------------------
*             oitmd - sub-outer iteration count                           40        terminates sub-outers
*             eps   - diffusion lambda-1.0 (see note below)            1.00E-03     does another sub-outer
*
*             note: eps, when the problem is finally converged, will equal epsi, the value shown above. however,
*                   early in the iteration process, a larger value may be used to avoid unnecessary iterations.
*
*                                            final convergence criteria
*
*          criterion    quantity to test                                  value     action taken if value exceeded
*          ---------    ----------------                                  -----     ------------------------------
*             oitm  - outer iteration count                               20        quits with error message
*             epsi  - transport lambda-1.0                             1.00E-03     does another outer
*
*******************************************************************************************************************************
*******************************************************************************************************************************

*******************************************************************************************************************************
*******************************************************************************************************************************
*
*                    ...flux and eigenvalue convergence as monitored by twodant...
*
*******************************************************************************************************************************
*****************************
*key start iteration monitor *
*****************************
*
*
* cpu time outer       diffusion                                  max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  sourc multip lambda-1          flux change  fiss change converged
* 410.45     0
*
*    ---------------------------------------------------------------------
*
*              - inner iteration summary for outer iteration no.   1 -
*
*                                  iter per max flux   at
*                                group  group   change  mesh
*                                  1     9   0.56E-03  25, 23
*
*
* cpu time outer       diffusion                                  max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  sourc multip lambda-1          flux change  fiss change converged
* 414.65     1      9       0      1.00000000 -1.00000E+00      5.64421E-04  0.00000E+00   yes
*
*               $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  0.00000E+00     total inners all outers =    9
*
*******************************************************************************************************************************
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                        ...sitno  1 link monitor...
*
*****************************************************************************************************************************************
*
*
*  the current fit coefficients are:
*  3.628E-02
*
*  number of histories, sitno  1 link calculation =      83750
*  cpu time required =  1.070E+02 sec
*  avg. no. of coll/hist =   3.94
*  avg. source particle weight =   0.9633
*  min source particle weight  =   0.0666
*  max source particle weight  =   0.9992
*  weight balance = -6.661E-16
*
*
*  particles generated through weight splitting   =          0
*  weight gain =  0.000E+00
*  weight loss =  0.000E+00
*
*  the maximum bank number was      0
*
*  the iteration error in the outgoing boundary flux is 3.90E-01     g =  1     i =   40
*  the iteration error in the mc to sn source is        5.82E-01     g =  2     k =  280
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************

*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                      ...flux and eigenvalue convergence as monitored by twodant...
*
*****************************************************************************************************************************************
*******************************
*key start iteration monitor *
*******************************
*
*
* cpu time outer       diffusion                                 max ptwise   max ptwise   inners
*   (sec)    no. inners sub-outers  sourc multip lambda-1        flux change  fiss change converged
* 522.33     0
*
*    ------------------------------------------------------------
*
*             -- inner iteration summary for outer iteration no.   1 --
*
*                                iter per max flux   at
*                             group  group   change   mesh
*                               1      8   0.33E-03  22, 40
*
*
* cpu time outer       diffusion                                 max ptwise   max ptwise   inners
*   (sec)    no. inners sub-outers  sourc multip lambda-1        flux change  fiss change converged
* 526.02     1      8       0      1.00000000 -1.00000E+00     3.32384E-04  0.00000E+00   yes
*
*               $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  0.00000E+00      total inners all outers =     8
*
*****************************************************************************************************************************************
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                        ...sitno  2 link monitor...
*
*****************************************************************************************************************************************
*
*  the fit error in the incoming boundary flux is 1.64E-01     g =  1     i =   39
*  the fit error in the sn to mc source is        0.00E+00     g =  0     k =    0
*
*  the current fit coefficients are:
*  4.892E-02  2.855E-03
*
*  number of histories, sitno  2 link calculation =     11125
*  cpu time required =  1.445E+01 sec
*  avg. no. of coll/hist =   4.02
*  avg. source particle weight =   0.6966
*  min source particle weight  =   0.0091
*  max source particle weight  =   0.9992
*  weight balance =  3.109E-15
*
*
*  particles generated through weight splitting   =         0
*  weight gain =  0.000E+00
*  weight loss =  0.000E+00
*
*  the maximum bank number was     0
*
*  the iteration error in the outgoing boundary flux is 1.86E-01     g =  1     i =   38
*  the iteration error in the mc to sn source is        2.55E-01     g =  2     k =  280
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************

*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                     ...flux and eigenvalue convergence as monitored by twodant...
*
*****************************************************************************************************************************************
*****************************
*key start iteration monitor *
*****************************
*
*
* cpu time outer       diffusion                                max ptwise   max ptwise   inners
*  (sec)     no. inners sub-outers  sourc multip lambda-1      flux change  fiss change converged
* 541.18     0
*
*     ------------------------------------------------------------
*
*              -- inner iteration summary for outer iteration no.   1 --
*
*                                 iter per max flux   at
*                            group  group   change   mesh
*                              1     7   0.66E-03  25, 40
*
*
* cpu time outer       diffusion                                max ptwise   max ptwise   inners
*  (sec)     no. inners sub-outers  sourc multip lambda-1      flux change  fiss change converged
* 544.78     1      7       0      1.00000000 -1.00000E+00  6.63192E-04  0.00000E+00   yes
*
*                $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  0.00000E+00     total inners all outers =    7
*
*****************************************************************************************************************************************
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                        ...sitno  3 link monitor...
*
**********************************************************************************************************************************
*
*  the fit error in the incoming boundary flux is 3.15E-02      g =  1     i =   36
*  the fit error in the sn to mc source is        0.00E+00      g =  0     k =    0
*
*  the current fit coefficients are:
*  5.358E-02  4.661E-03  2.812E-04
*
*  number of histories, sitno  3 link calculation =       6000
*  cpu time required =  7.933E+00 sec
*  avg. no. of coll/hist =    3.90
*  avg. source particle weight =    0.1375
*  min source particle weight  =    0.0016
*  max source particle weight  =    0.6210
*  weight balance = -5.107E-15
*
*
*  particles generated through weight splitting   =          0
*  weight gain =  0.000E+00
*  weight loss =  0.000E+00
*
*  the maximum bank number was      0
*
*  the iteration error in the outgoing boundary flux is 9.24E-02      g =  1     i =   38
*  the iteration error in the mc to sn source is        1.29E-01      g =  2     k =  280
*
**********************************************************************************************************************************
**********************************************************************************************************************************

**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                      ...flux and eigenvalue convergence as monitored by twodant...
*
**********************************************************************************************************************************
*******************************
*key start iteration monitor *
*******************************
*
*
* cpu time outer       diffusion                              max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  sourc multip lambda-1      flux change  fiss change converged
* 553.45     0
*
*     ------------------------------------------------------------                    ------
*
*              -- inner iteration summary for outer iteration no.   1 --
*
*                                  iter per max flux   at
*                              group  group   change   mesh
*                                1     6   0.68E-03  25, 40
*
*
* cpu time outer       diffusion                              max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  sourc multip lambda-1      flux change  fiss change converged
* 556.25     1      6       0      1.00000000 -1.00000E+00  6.78981E-04  0.00000E+00   yes
*
*                $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  0.00000E+00      total inners all outers =     6
*
**********************************************************************************************************************************
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                        ...sitno  4 link monitor...
*
*****************************************************************************************************************************
*
*  the fit error in the incoming boundary flux is 5.92E-03     g =  1     i =   40
*  the fit error in the sn to mc source is        0.00E+00     g =  0     k =    0
*
*  the current fit coefficients are:
*  5.534E-02  5.569E-03  5.315E-04  3.152E-05
*
*  number of histories, sitno  4 link calculation =      6000
*  cpu time required =  7.817E+00 sec
*  avg. no. of coll/hist =   3.91
*  avg. source particle weight =   0.0125
*  min source particle weight  =   0.0001
*  max source particle weight  =   0.0544
*  weight balance = -6.661E-15
*
*
*  particles generated through weight splitting   =         0
*  weight gain =  0.000E+00
*  weight loss =  0.000E+00
*
*  the maximum bank number was     0
*
*  the iteration error in the outgoing boundary flux is 4.65E-02     g =  1     i =   38
*  the iteration error in the mc to sn source is        6.69E-02     g =  2     k =  280
*
*****************************************************************************************************************************
*****************************************************************************************************************************

*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                     ...flux and eigenvalue convergence as monitored by twodant...
*
*****************************************************************************************************************************
*******************************
*key start iteration monitor *
*******************************
*
*
* cpu time outer       diffusion                                 max ptwise   max ptwise   inners
*  (sec)     no. inners sub-outers  sourc multip lambda-1        flux change  fiss change converged
* 564.63      0
*
*     ------------------------------------------------------------
*
*              — inner iteration summary for outer iteration no.   1 —
*
*                                  iter per max flux   at
*                               group  group   change   mesh
*                                 1     5   0.76E-03  22, 40
*
*
* cpu time outer       diffusion                                 max ptwise   max ptwise   inners
*  (sec)     no. inners sub-outers  sourc multip lambda-1        flux change  fiss change converged
* 567.00      1      5       0      1.00000000 -1.00000E+00     7.55332E-04  0.00000E+00   yes
*
*                $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  0.00000E+00      total inners all outers =    5
*
*****************************************************************************************************************************
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                        ...sitno  5 link monitor...
*
**********************************************************************************************************************************
*
*  the fit error in the incoming boundary flux is 8.92E-04     g =  1     i =  47
*  the fit error in the sn to mc source is        0.00E+00     g =  0     k =   0
*
*  the current fit coefficients are:
*  5.603E-02  5.990E-03  6.890E-04  6.611E-05
*
*  the projected fit coefficients are:
*  5.650E-02  6.328E-03  8.584E-04  1.213E-04
*
*  the iteration error in the outgoing boundary flux is 4.33E-02     g =  1     i =  38
*  the iteration error in the mc to sn source is        6.46E-02     g =  2     k = 280
*
**********************************************************************************************************************************
**********************************************************************************************************************************

**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                        ...flux and eigenvalue convergence as monitored by twodant...
*
**********************************************************************************************************************************
*******************************
*key start iteration monitor *
*******************************
*
*
* cpu time outer       diffusion                                  max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  sourc multip lambda-1          flux change  fiss change converged
* 567.22     0
*
*     ------------------------------------------------------------
*
*              -- inner iteration summary for outer iteration no.   1 --
*
*                                  iter per max flux   at
*                              group  group   change   mesh
*                                1      5   0.70E-03  22, 40
*                                2      9   0.85E-03   6, 39
*
*
* cpu time outer       diffusion                                  max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  sourc multip lambda-1          flux change  fiss change converged
* 573.33     1     14       0     1.00000000 -1.00000E+00       8.46207E-04  0.00000E+00   yes
*
*                $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  1.27089E-05     total inners all outers =   14
*
**********************************************************************************************************************************
```

```
*******************************************************************************************************************************
*******************************************************************************************************************************
*
*                                        ...sitno  6 link monitor...
*
*******************************************************************************************************************************
*
*  the fit error in the incoming boundary flux is 3.31E-03     g =  1     i =   47
*  the fit error in the sn to mc source is        0.00E+00     g =  0     k =    0
*
*  the current fit coefficients are:
*  5.650E-02  6.328E-03  8.580E-04  1.214E-04
*
*  the iteration error in the outgoing boundary flux is 1.51E-05     g =  1     i =   36
*  the iteration error in the mc to sn source is        2.59E-05     g =  2     k =  245
*
*******************************************************************************************************************************
*******************************************************************************************************************************

*******************************************************************************************************************************
*******************************************************************************************************************************
*
*                                        ... monte carlo region particle balance ...
*
*******************************************************************************************************************************
*
*    gp          source        sn source       fission src     net leakage      absorption
*
*     1       1.0000E+00       0.0000E+00       0.0000E+00       7.5904E-01       8.0316E-02
*     2       0.0000E+00       0.0000E+00       0.0000E+00       0.0000E+00       0.0000E+00
*   tot       1.0000E+00       0.0000E+00       0.0000E+00       7.5904E-01       8.0316E-02
*
*
*    gp        in scatter     self scatter     out scatter   particle balance
*
*     1       0.0000E+00       6.0237E+00       1.6061E-01       3.0391E-05
*     2      -1.5238E-17       0.0000E+00       0.0000E+00       0.0000E+00
*   tot      -1.5238E-17       6.0237E+00       1.6061E-01       3.0391E-05
*
*******************************************************************************************************************************
*******************************************************************************************************************************

*******************************************************************************************************************************
*******************************************************************************************************************************
*
*                                        mc coarse mesh leakages
*
*******************************************************************************************************************************
*
*                                          top   leakages
*
*        i       1   i       2   i       3
*       1.6261E-02  2.2988E-03  0.0000E+00
*         0.0191      0.0452      0.0000
*
*                                        bottom leakages
*
*        i       1   i       2   i       3
*       1.3818E-01  9.8726E-02  0.0000E+00
*         0.0049      0.0056      0.0000
*
*                                        right leakages
*
*        j       1
*       1.0477E+00
*         0.0009
*
*                                        left  leakages
*
*        j       1
*       0.0000E+00
*         0.0000
*
*******************************************************************************************************************************
*******************************************************************************************************************************

*******************************************************************************************************************************
*******************************************************************************************************************************
*
*                                        tracks entering (coarse mesh, zm calc)
*
*******************************************************************************************************************************
*
*        i       1   i       2   i       3
*    1       51507       152294           0
*
*******************************************************************************************************************************
*******************************************************************************************************************************
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                        ...group edit and balances upon convergence...
*
**********************************************************************************************************************************
*
*
*                    ...title---  sample sn/mc problem for twodant user's guide                    ...
*
*
*                                        ...system balance tables... (neutrons only)
*
*****************************
*key start balance table    *
*****************************
*
*
*       gp          source      fission source         in scatter        self scatter       out scatter        net leakage
*
*        1  1.0000000E+00       0.0000000E+00      -2.2204460E-16       1.6381455E+01      4.1635716E-01       3.7543551E-01
*        2  0.0000000E+00       0.0000000E+00       4.1635716E-01       2.1997356E+00      2.2204460E-15       2.5514699E-02
*
*      tot  1.0000000E+00       0.0000000E+00       4.1635716E-01       1.8581191E+01      4.1635716E-01       4.0095021E-01
*
*
*
*       gp       absorption   particle balance      right leakage  horizontal leakage       top leakage    vertical leakage
*
*        1  2.0818928E-01       1.8054936E-05       9.4993408E-03       9.4993408E-03      2.0443840E-02       3.6593617E-01
*        2  3.9084251E-01      -1.3126612E-07       8.5466375E-04       8.5466375E-04      8.6766648E-04       2.4660035E-02
*
*      tot  5.9903179E-01       1.2708858E-05       1.0354005E-02       1.0354005E-02      2.1311506E-02       3.9059620E-01
*
*
*
*       gp     left leakage    bottom leakage      nprod spectrum
*
*        1  0.0000000E+00       3.4549233E-01       0.0000000E+00
*        2  0.0000000E+00       2.3792369E-02       0.0000000E+00
*
*      tot  0.0000000E+00       3.6928470E-01       0.0000000E+00
*
*    Timing for sweeper and DSA...tsweep=  13.933 and tdsa=   6.367 and tgrey=   0.000 seconds.
*
*
**********************************************************************************************************************************

integral summary information
summary eigenvalue not used
integral-source-i          neutron  1.0000000E+00
integral-fission-i         neutron  0.0000000E+00
integral-in-scak-i         neutron  4.1635716E-01
integral-self-scak-i       neutron  1.8581191E+01
integral-out-scak-i        neutron  4.1635716E-01
integral-net lkage-i       neutron  4.0095021E-01
integral-absorption-i      neutron  5.9903179E-01
integral-right lkage-i     neutron  1.0354005E-02
integral-horizontal lkage-i neutron 1.0354005E-02
integral-top lkage-i       neutron  2.1311506E-02
integral-vertical lkage-i  neutron  3.9059620E-01
```

```
**********************************************************************************************************************************************
**********************************************************************************************************************************************
************************************
*key start coarse mesh balances    *
************************************
*
*
*       definition:
*
*            balance =  1.0  -  ------------------------------------------------------------------------------------------
*                               q(fixed) + q(fission) + inscat + a*j+(left) + a*j-(right) + a*j+(bottom) + a*j-(top)
*
*                       note: boundary sources are included in the currents
*
****************************************** coarse mesh balances for group   1 ************************************************
*  mesh volumetric    fission     in-       out-                 effective  left j+    left j-    bottom j+  bottom j-  particle
* (i,j) fixed source   source    scatter    scatter  absorption absorption times area times area times area times area  balance
*  1  1 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 1.0000E+00 1.3818E-01 2.7917E-12
*  2  1 0.0000E+00 0.0000E+00 0.0000E+00 1.6061E-01 8.0316E-02 8.0316E-02 1.7400E+00 8.9445E-01 0.0000E+00 9.8726E-02 1.0743E-05
*  3  1 0.0000E+00 0.0000E+00 0.0000E+00 2.5575E-01 1.2787E-01 1.2787E-01 1.0477E+00 5.4415E-01 0.0000E+00 1.0859E-01-6.1885E-06
*  4  1                                                                   9.4993E-03-3.4694E-18
*  1  2                                                                                         1.6261E-02 0.0000E+00
*  2  2                                                                                         2.2989E-03 0.0000E+00
*  3  2                                                                                         1.8841E-03 0.0000E+00
* total 0.0000E+00 0.0000E+00 0.0000E+00 4.1636E-01 2.0819E-01 2.0819E-01 0.0000E+00 0.0000E+00 1.0000E+00 3.4549E-01 1.8055E-05
*                                                                        9.4993E-03-3.4694E-18 2.0444E-02 0.0000E+00
*
****************************************** coarse mesh balances for group   2 ************************************************
*  mesh volumetric    fission     in-       out-                 effective  left j+    left j-    bottom j+  bottom j-  particle
* (i,j) fixed source   source    scatter    scatter  absorption absorption times area times area times area times area  balance
*  1  1 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 8.5278E-03 4.4409E-16
*  2  1 0.0000E+00 0.0000E+00 1.6061E-01-6.4643E-17 1.2810E-01 1.2810E-01 5.9699E-02 6.8774E-02 0.0000E+00 6.2638E-03-5.2901E-07
*  3  1 0.0000E+00 0.0000E+00 2.5575E-01 3.2412E-16 2.6275E-01 2.6275E-01 7.9703E-02 6.2676E-02 0.0000E+00 9.0001E-03 1.5137E-06
*  4  1                                                                   8.5466E-04-2.1684E-19
*  1  2                                                                                         5.4720E-04 0.0000E+00
*  2  2                                                                                         1.4968E-04 0.0000E+00
*  3  2                                                                                         1.7101E-04 0.0000E+00
* total 0.0000E+00 0.0000E+00 4.1636E-01 2.5948E-16 3.9084E-01 3.9084E-01 0.0000E+00 0.0000E+00 0.0000E+00 2.3792E-02 8.6002E-07
*                                                                        8.5466E-04-2.1684E-19 8.6790E-04 0.0000E+00
*
****************************************** coarse mesh balances summed over groups ********************************************
*  mesh volumetric    fission     in-       out-                 effective  left j+    left j-    bottom j+  bottom j-  particle
* (i,j) fixed source   source    scatter    scatter  absorption absorption times area times area times area times area  balance
*  1  1 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 1.0000E+00 1.4671E-01 2.6938E-12
*  2  1 0.0000E+00 0.0000E+00 1.6061E-01 1.6061E-01 2.0841E-01 2.0841E-01 1.7997E+00 9.6323E-01 0.0000E+00 1.0499E-01 9.5005E-06
*  3  1 0.0000E+00 0.0000E+00 2.5575E-01 2.5575E-01 3.9062E-01 3.9062E-01 1.1274E+00 6.0683E-01 0.0000E+00 1.1759E-01-4.3206E-06
*  4  1                                                                   1.0354E-02-3.6863E-18
*  1  2                                                                                         1.6808E-02 0.0000E+00
*  2  2                                                                                         2.4485E-03 0.0000E+00
*  3  2                                                                                         2.0551E-03 0.0000E+00
* total 0.0000E+00 0.0000E+00 4.1636E-01 4.1636E-01 5.9903E-01 5.9903E-01 0.0000E+00 0.0000E+00 1.0000E+00 3.6928E-01 1.3000E-05
*                                                                        1.0354E-02-3.6863E-18 2.1312E-02 0.0000E+00
*
*
*                                        ...interface file rtflux written..
*
*
*                                        ...interface file sncons written..


twodant iteration time, mins  9.5611E+00
```

```
*********************************************************************************************************
*********************************************************************************************************
*
*
*                                        run highlights
*******************************
*key start run highlights     *
*******************************
*                                   ****************************************
*                                   *                                      *
*                                   *     all modules are tentatively go.  *
*                                   *      interface file geodst written.  *
*                                   *         cross sections from cards.   *
*                                   *     interface mixing files written.  *
*                                   *      interface file asgmat written   *
*                                   *     xs files macrxs,snxedt written.  *
*                                   *      interface file solinp written.  *
*                                   *    edit module execution suppressed. *
*                                   *   neither editit nor edit cards exist.*
*                                   *         start solver execution.      *
*                                   *      all convergence criteria met.   *
*                                   *      all convergence criteria met.   *
*                                   *      all convergence criteria met.   *
*                                   *      all convergence criteria met.   *
*                                   *      all convergence criteria met.   *
*                                   *      all convergence criteria met.   *
*                                   *      interface file rtflux written.  *
*                                   *      interface file sncons written.  *
*                                   *                                      *
*                                   ****************************************
*
*
*                                       storage and timing history
*
*                         **************************************************************
*                         *          scm      scm      lcm      lcm      cpu      sysio *
*                         * module  words    limit    words    limit   seconds  seconds *
*                         * ------  -----    -----    -----    -----   -------  ------- *
*                         *    0        0        0        0        0    573.7      0.0  *
*                         *  100    11036    40000        0        0      2.0      0.0  *
*                         *  101        0        0        0        0      0.7      0.0  *
*                         *  102     1041    40000        5   140000      0.0      0.0  *
*                         *  103       39    40000        0        0      0.0      0.0  *
*                         *  104     2032    40000        0        0      0.0      0.0  *
*                         *  105        0        0        0        0      0.0      0.0  *
*                         *  106        0        0        0        0      0.0      0.0  *
*                         *  107      107    40000       48   140000      0.1      0.0  *
*                         *  108      651    40000       36   140000      0.1      0.0  *
*                         *  109        0        0        0        0      0.0      0.0  *
*                         *  112        0        0        0        0      0.0      0.0  *
*                         *  200    16000    40000    68478   140000    571.7      0.0  *
*                         *  201        0        0        0        0      0.4      0.0  *
*                         *  202        0        0        0        0      0.1      0.0  *
*                         *  203     1204    40000        0        0      0.0      0.0  *
*                         *  204        0        0        0        0      0.0      0.0  *
*                         *  205        0        0        0        0    408.0      0.0  *
*                         *  206        0        0        0        0     22.4      0.0  *
*                         *  207        0        0        0        0      0.0      0.0  *
*                         *  208        0        0        0        0      0.4      0.0  *
*                         *  210        0        0        0        0      0.1      0.0  *
*                         *  211        0        0        0        0      0.0      0.0  *
*                         *  400        0        0        0        0      0.0      0.0  *
*                         **************************************************************
*
*
*                                    ...execution terminated...
*
*
*********************************************************************************************************
*********************************************************************************************************
```

APPENDIX B: OPERATING SYSTEM SPECIFICS

UNIX/UNICOS Execution

On UNIX or UNICOS systems, the input is on STDIN and the printed output is on STDOUT. Thus, the user will normally cause execution of the program with the command:

dant.x < *odninp* > *odnout*

where dant.x is the name of the executable file, *odninp* is the user's choice for a name for the input file, and *odnout* is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is dant.x.

STDERR contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDERR are any error messages.

Library Search Path

Most files read or written by TWODANT are in the current UNIX working directory. Some forms of cross-section files may be kept in other directories. By setting the environment variable SNXSPATH, the user may specify an ordered set of alternate directories in which the program should look for the named files. As an example, if an ISOTXS file is in the directory, `/usr/tmp/xs`, then the following command can be used

```
setenv SNXSPATH /usr/tmp/xs
```

and TWODANT will then look in that named directory for the library. The search path for each of the possible libraries is given in Table 3.2.

Table 3.2 UNIX Search Path

LIB	SEARCH PATH
MACRXS	Current Working Directory (CWD).
GRUPXS	**SNXSPATH**, then CWD.
ISOTXS	**SNXSPATH**, then CWD.
BXSLIB	**SNXSPATH**, then CWD, but see text below.
ODNINP	None, the library is contained in the input file.
MACBCD	CWD
XSLIBB	CWD
MENDF[a]	Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.
MENDFG[b]	
XSLIB	**SNXSPATH**, then CWD
other	For any name other than those above, the program will assume the form is XSLIB and search for it in **SNXSPATH**, then CWD.

a. Available only at Los Alamos.

b. Available only at Los Alamos.

SNXSPATH can be used to protect an input BXSLIB file from being overwritten. See the discussion on page 3-40.

TWODANT/GQ USER'S GUIDE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

4

XTM — Transport Methods Group

Los Alamos
National Laboratory

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

An Affirmative Action/Equal Opportunity Employer

This work was supported by the US Department of Energy.

USER'S GUIDE FOR TWODANT/GQ: A CODE PACKAGE FOR TWO-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT USING GENERALIZED QUADRILATERAL MESHES

by
Ray E. Alcouffe, Forrest W. Brinkley,
and Duane R. Marr

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The TWODANT/GQ code package is a modular computer program designed to solve the two-dimensional, time-independent, multigroup discrete-ordinates form of the Boltzmann transport equation[1,2] on a generalized quadrilateral mesh.

TWODANT/GQ is based on the modular construction of the DANTSYS™ package. This modular construction separates the input processing, the transport equation solving, and the postprocessing, 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 used by all the codes in the package. With these modules, production codes with different solution techniques are invoked simply by executing different SOLVER modules in the package. This SOLVER choice is automatically made by the package through an analysis of the input stream.

The TWODANT/GQ code is then simply the DANTSYS package with a two-dimensional SOLVER module with the capability of handling a generalized quadrilateral mesh.

Some of the major features included in the DANTSYS package are:

1. a free-field format ASCII text input capability, designed with the user in mind,
2. standardized data and file management techniques as defined[3] and developed by the Committee on Computer Code Coordination (CCCC); both sequential file and random-access file handling techniques are used,
3. the use of a diffusion synthetic acceleration scheme[4] to accelerate the iterative process in the SOLVER module,
4. direct (forward) or adjoint calculational capability,
5. x-y and r-z geometry options,
6. arbitrary anisotropic scattering order,
7. vacuum, reflective, periodic, translational, rotational, white, or surface source boundary condition options on arbitrary faces,
8. inhomogeneous (fixed) source or k_{eff} calculation options, as well as time-absorption (alpha) and nuclide concentration search options

---

DANTSYS and TWODANT/GQ are trademarks of the Regents of the University of California, Los Alamos National Laboratory.

9. "diamond-differencing" for solution of the transport equation,
10. a diffusion solver that uses the multigrid method,[5]
11. user flexibility in using either ASCII text or sequential file input,
12. user flexibility in controlling the execution of both modules and submodules, and
13. extensive, user-oriented error diagnostics.

DOCUMENTATION FOR TWODANT/GQ USAGE

The documentation described here constitutes a complete manual for the use of the TWODANT/GQ code.

Included are two general categories of information. The first category is in this User's Guide and is oriented towards preparing input to the code. The second category is of a background, reference, conceptual, or theoretical nature and is intended primarily for the novice or first time user; an experienced user generally needs only this User's Guide.

What Is In This User's Guide

This User's Guide is a chapter from the much larger DANTSYS document. This Guide provides the ASCII text input specifications for TWODANT/GQ.

The guide is intended to serve as a complete input manual for two classes of user. Special, succinct sections containing summaries and compact tables are intended for the advanced user in order to make his input preparation more efficient. The main body of the guide concerns itself with descriptions of the input and should be sufficient for the user familiar with discrete ordinates concepts. Novice users may find other chapters of the document necessary.

This Guide first gives an overview of the input block order required by the code.

Next is a "mini-manual" in which are listed all the names of available input arrays arranged by input block. Definitions of input arrays are not given, as the names are suggestive, but expected types and sizes are provided. This mini-manual is very useful to the user as a quick check for completeness, a quick reference to type and size, and as an index into the more detailed array descriptions that follow. For the experienced user, the mini-manual is frequently all that is needed to prepare a complete input deck.

Following the mini-manual are reference sections describing in detail all the input parameters and arrays.

Appendix A provides a sample TWODANT/GQ case with model, input, and output descriptions.

Lastly, Appendix B details operating system specifics, including how to effect an execution of the code.

Information of a reference, background, or theoretical nature that the first time user may need may not be found in this User's Guide, but the user will encounter liberal references to other chapters of this document for that sort of information.

What Is Available Elsewhere

In addition to this User's Guide, the user, especially the first time user, may find the information below described in other chapters of this document pertinent. For even greater detail on some of the general items, particularly the methods items, the user should look at Ref. 6.

The chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1 discusses in more detail the geometry and solver concepts and their related input. If the User's Guide proves insufficient for your needs, look in this chapter. Among the many sections of the chapter are ones on the input of inhomogeneous sources and a discussion of eigenvalue searches. There is also more detail on the Block-I input.

A discussion of how the EDIT module works and more detail on preparing the input is given in the chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 serves as the reference manual for the free-field input (rules, format, and operators) used in this code. That chapter is summarized in this guide, but should the summary prove inadequate, the user is referred there for full details.

The chapter "CROSS-SECTION LIBRARIES" starting on page 10-1 gives details of the many library formats available to TWODANT/GQ, including sections on how to prepare your own card-image (or text) libraries.

The chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1 describes the mixing concepts in detail and shows some examples.

Next is the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. That chapter describes the theoretical basis for the TWODANT/GQ code as well as the other codes in the DANT-SYS package.

In the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREED-ANT — Code Structure" starting on page 13-1 is shown a brief 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. In particular, the code package structure must be understood in order to make up input for piecewise executions of the code that are possible with controls that are part of the input in Block-I.

Error diagnostics that the user might encounter are found in the chapter "ERROR MES-SAGES" starting on page 14-1. Several examples of input errors and the resulting error messages are provided for the user.

The chapter "FILE DESCRIPTIONS" starting on page 15-1 is a reference that describes all the files used by the package. Included is a detailed description of the file structure of the code dependent, binary, sequential interface files generated by and used in the

DANTSYS package. Also included are descriptions of any other files produced or used by the package, both binary and text. In some cases, this may simply be a reference to a more comprehensive document, such as the file descriptions for the CCCC standard interface files.

GEOMETRY CONCEPTS

Discrete ordinates codes commonly use a rectilinear mesh, that is, one in which every mesh interval is a rectangle. It follows that every interior node or mesh vertex is connected to four other nodes and only on the periphery of the mesh may there be two or three node connectivity.

TWODANT/GQ uses a generalized quadrilateral mesh that has the same connectivity features as the rectilinear mesh, but relaxes the restriction that each mesh interval must be a rectangle, thus allowing nonvertical and nonhorizontal lines. This allows the user to much more accurately model complex geometries.

TWODANT/GQ also allows the user to construct an elaborate geometry from smaller pieces. As a simple example, one could specify the mesh for a lattice cell and then specify many invocations of this cell to form the mesh for a complete lattice assembly. The mesh for such a lattice cell is termed a "submesh."

Submeshes are combined to make the complete mesh by tiling. For TWODANT/GQ, there must be no gaps between submeshes in the full mesh nor other holes of any kind in the full mesh, and vertexes must match on the interfaces between submeshes.

For a complete geometry description, one needs to specify not only the submeshes but also their content and how they fit together. Consider the following definitions of geometric entities, after which we shall present examples of each as we construct a composite geometry.

Definition: A submesh is a purely geometric construct of vertexes and the connecting edges that define mesh intervals. It contains no materials as yet.

Definition: A zoning is the specification of a zone number for each of the intervals in a submesh. The zone numbers will later be correlated with materials.

Definition: An object is a submesh with content, that is, a submesh together with a zoning (loading).

Definition: A component is a composite of one or more objects, each of which has position and orientation.

Definition: The geometry is a composite of one or more components (or objects) that describe the problem solution domain. Only the components described in the geometry are included in the solution domain.

Definition: A boundary condition description is the specification of a boundary segment together with the boundary condition applying upon it.

Submeshes

Some examples of submeshes that can be described by the TWODANT/GQ input are shown in Figure 4.1. Notice that each of these submeshes is logically rectangular, that is, it is formed from an m by n array of verticies. Also notice that each mesh interval is a quadrilateral.

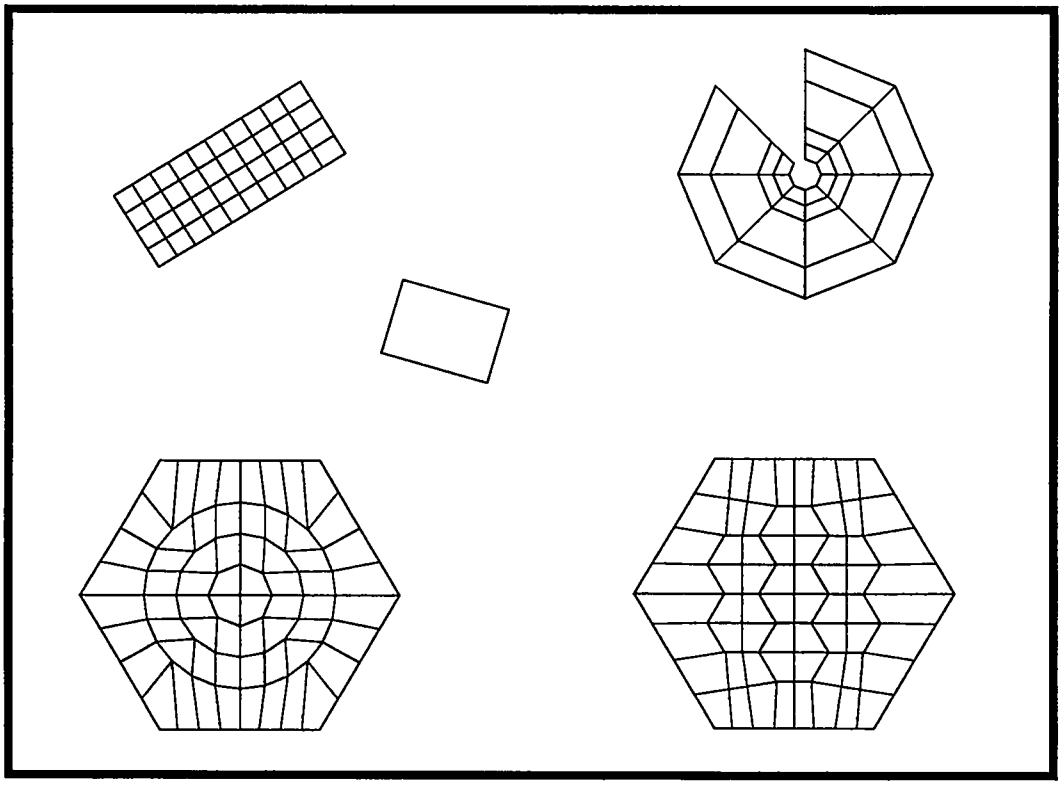


Figure 4.1 Submesh Examples

Objects

The zone number must be specified for each of the mesh intervals in the submesh. This logically rectangular m-1 by n-1 array of integers is termed a zoning; m and n are the number of verticies in, respectively, the x and y directions. Later, materials will be associated with each of the zone numbers, thus producing a submesh with material content. The combination of a submesh together with a suitable zoning is termed an object.

Three objects formed from two of the submeshes from above are shown in Figure 4.2. In the objects, different zones are shown with different shading. Notice that the same submesh can be used with two different zonings to produce two different objects.

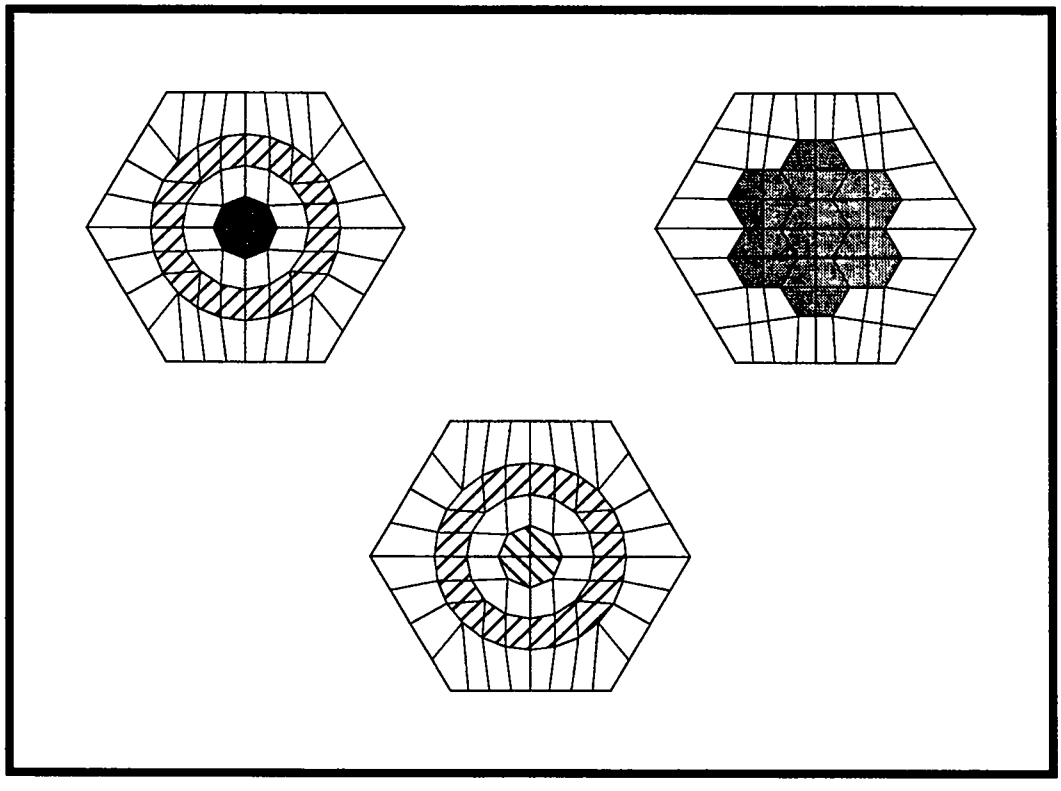


Figure 4.2 Examples of Objects

Components

Objects may combined together to form a component by specifying each object and its orientation and position with respect to the other objects in the component. A component formed from three instances of one of the objects shown above together with a single instance of another of the objects is shown in Figure 4.3.

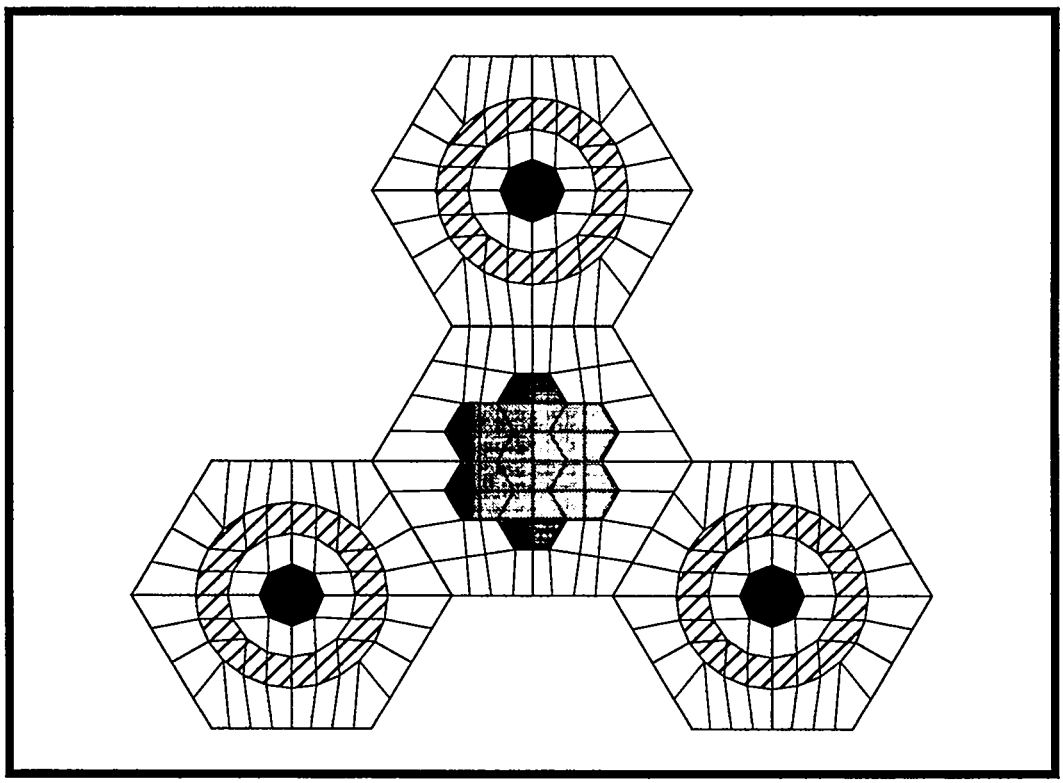


Figure 4.3 A Component Formed from Four Objects

Geometry

Finally, components and/or objects may be combined together to form the geometry. In Figure 4.4, we show the final geometry of a hypothetical seven subassembly space reactor. It is composed of the component formed above combined with three instances of one of the objects previously shown.

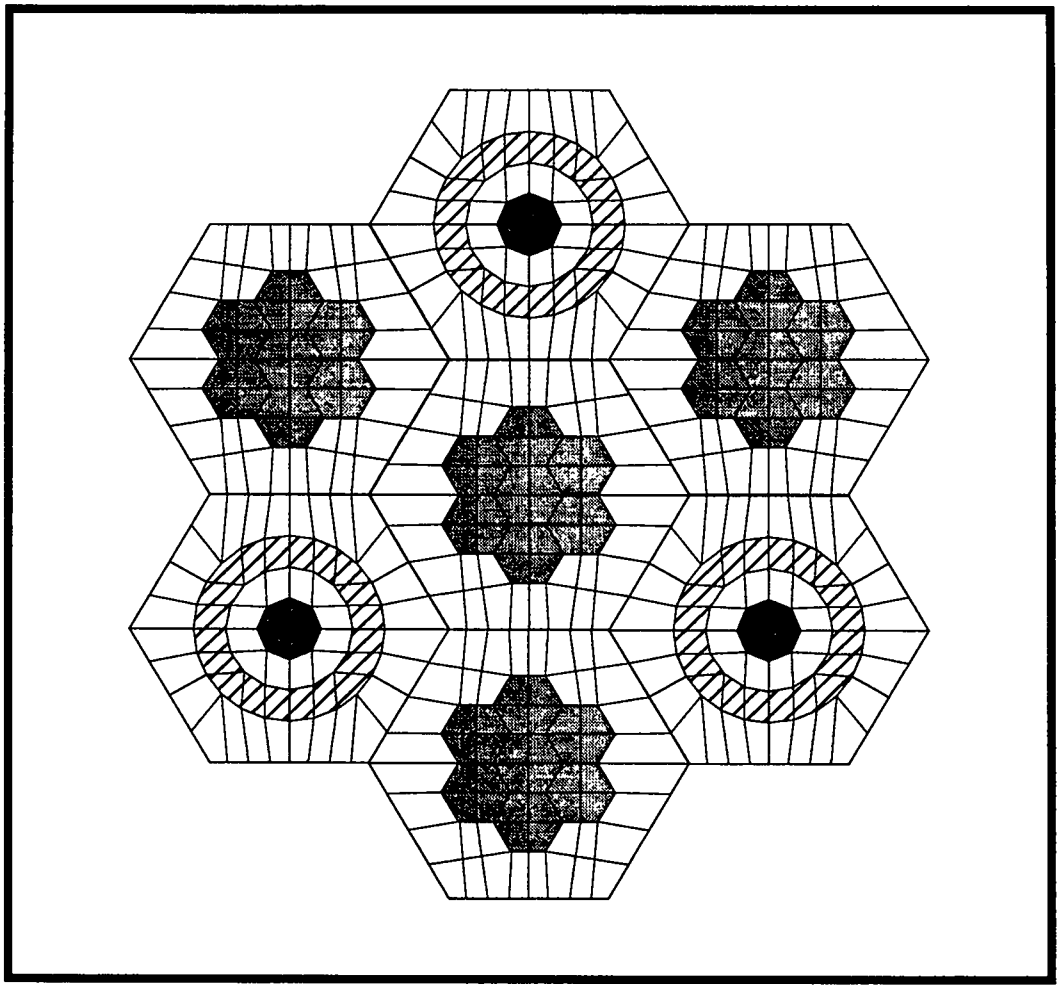


Figure 4.4 The Final Geometry

Another way to form this same geometry would be to skip the component definition step and specify all seven objects in the geometry specification. The geometry is really the highest level component in the formation tree but has the distinction of defining the problem solution domain. We chose, in our example here, to describe a separate component to illustrate the definition of a component as distinct from the geometry and to demonstrate that components can be nested within the formation tree.

Boundary Conditions

Boundary conditions may be specified in one of two ways. If the final geometry has the property of having a logically rectangular m by n mesh, that is, it has n rows of m mesh intervals and thus has (logical) left, right, top, and bottom edges, then one can specify the boundary condition along each of the logical edges.

In a more complex final geometry such as the one we have shown in Figure 4.4 (which is not logically rectangular), we have two alternatives. If there is vacuum all around, we

need do nothing as the code will default to vacuum. Otherwise, we must specify in detail what the boundary condition is all along the periphery of the final geometry.

Specifying the boundary condition applying on the periphery of a complex final geometry consists of specifying portions of the periphery, called boundary segments, together with the condition applying along each of those segments.

Figure 4.5 shows an example of a boundary segment running along the periphery of the geometry from vertex A to vertex B. It is shown slightly offset from the true periphery for illustrations purposes only. We specify the segment by specifying the coordinates of the vertices A and B. We need not specify any of the points in between.

Boundary segments have sense such that in traveling along the periphery from A to B, the interior is on the left and the exterior is on the right. In specifying the boundary segment, we must thus specify point A first and then point B. If we were to specify B first, then the segment being described would not be the one shown in the figure, but rather the rest of the periphery, that is, that portion starting at B and continuing around the periphery in a counter clockwise direction until arriving at point A.

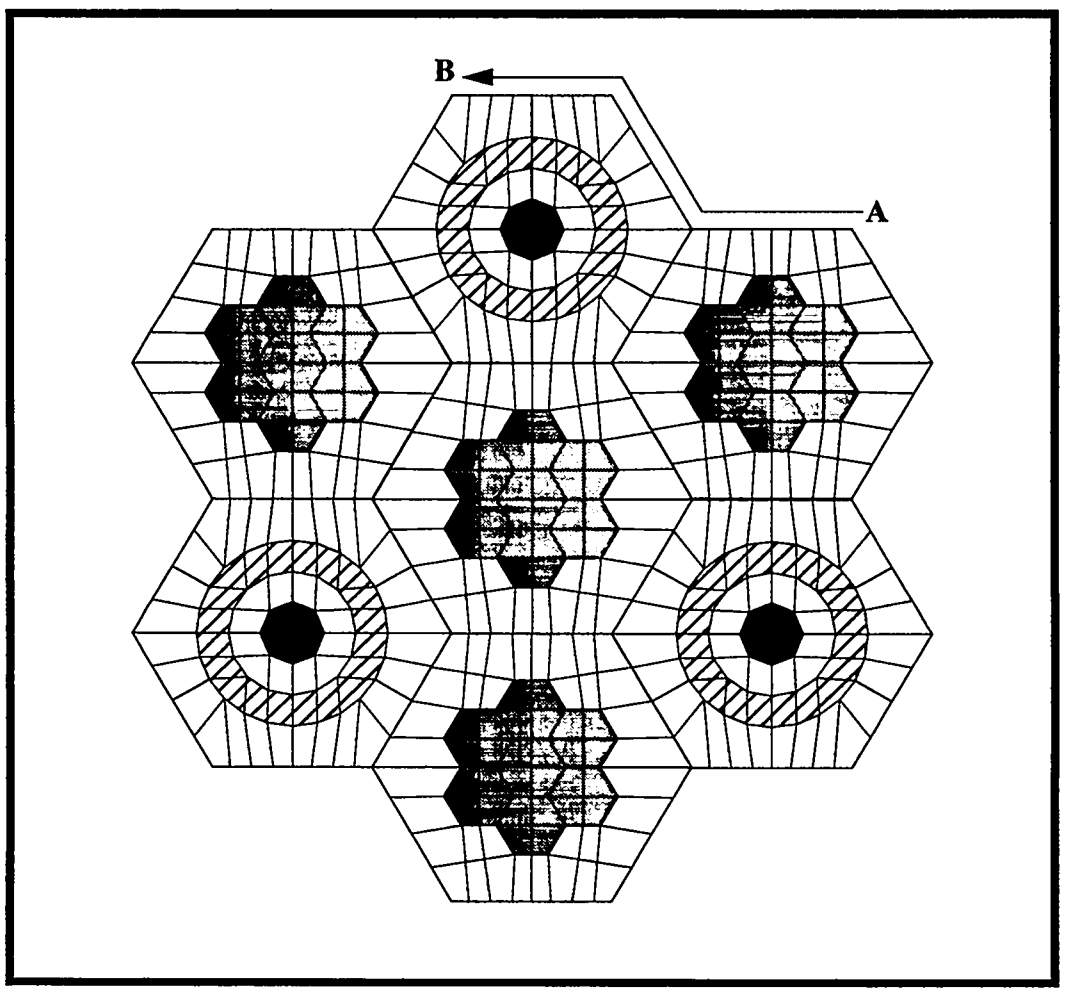


Figure 4.5 Example of a Boundary Segment

TWODANT/GQ INPUT OVERVIEW

Input Block Order

The full TWODANT/GQ input consists of a title line section, followed by six blocks of free field input. The title line section is not free field. Any input referred to as a block uses the free field input form.

Block-I consists of basic control and dimensional information that allows efficient packing of the array data. This information also allows checking of the lengths of arrays supplied in subsequent blocks or those from interface files.

Block-II contains the geometric information.

Block-III consists of the nuclear data specifications.

Block-IV contains mixing information.

Block-V contains the rest of the input needed for specifying the flux calculation.

And lastly, Block-VI contains the edit (i.e., report writing) specifications.

If a text cross-section library is to be included in the input deck, it should be placed between Blocks III and IV. TWODANT/GQ supports many library formats, and so the library may or may not be in free field format depending upon the option chosen.

A full input would then look like that diagrammed on the following page.

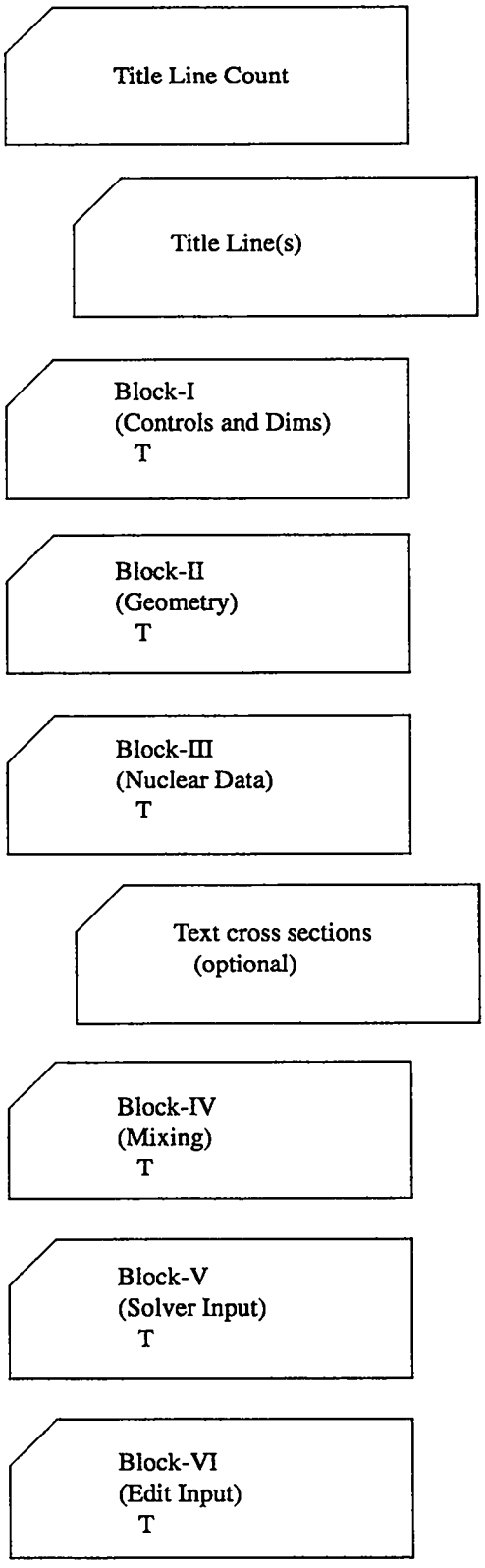


Figure 4.6 TWODANT/GQ Input Order

Free Field Input Summary

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 is summarized here for quick reference.

There are four basic input quantities in the free field input used in ONEDANT; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is briefly described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array." To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, might look like:

```
SOURCX= 0 0 0 1.1 1.1 0 0 0 0 0
```

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Single value input variables are treated as arrays of unit length.

Numeric Data Items

Numeric data items follow a Fortran input convention. For example, all of the following are valid entries for the number ten:

```
10, 1.0+1, 1E1, 10.0
```

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a Fortran variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters and blanks may be included if the data item is surrounded by double quotes. Operators may NOT be used with character data items.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this, the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described later, in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read, no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input.

The data operators are specified in the general form

n *O* d

where:

n is the "data numerator," either an integer or a blank;
O is any one of the "data operator" characters shown below; and
d is a "data entry" (may be blank for some operators).

Note: The "data operator" character must be appended to the "data numerator."

Using operators, the SOURCX input described above could more succinctly be given as:

SOURCX= 0 0 0 2R 1.1 F0

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO[7] operators may be used in numeric entry.

A table of the most used operators is given next including brief descriptions. For full descriptions of these and a complete list of all the available operators, including the more esoteric ones, the user is referred to "FREE FIELD INPUT DETAILS" on page 9-13.

Frequently Used Operators

Operator[a]	Functionality
nR d	REPEAT the data item d, n times.
nI d	INTERPOLATE (linear) n data items between data item d and the next data item.
nC d	SCALE (multiply) the n previous entries by d.
F d	FILL the rest of the data string with the data item d.
nY m	STRING REPEAT. Repeat the previous m strings, n times.
nL d	INTERPOLATE LOGARITHMICALLY n data items between d and the next d.
nZ	ZERO. Enter the value zero n successive times.
nS	SKIP. Skip the next n data items.
nQ m	SEQUENCE REPEAT. Enter the last m entries, n more times.
nG m	SEQUENCE REPEAT WITH SIGN CHANGE. Same as the Q option but the sign of the m entries is changed every repeat.
nN m	SEQUENCE REPEAT INVERT. Same as the Q option but the order of the m entries is inverted each repeat.
nM m	SEQUENCE REPEAT INVERT WITH SIGN CHANGE. Same as N option but the sign is also changed every repeat.
nX	COUNT CHECK. Causes code to check the number of entries in the current string so far, against the number n.

a. The operator character must always be appended directly to n. d or m need not be immediately adjacent to the operator character.

MINI-MANUAL Introduction

On the following few pages is given a complete list of the input names, expected array sizes, and order within the array. No description of the array contents is given in this MINI-MANUAL as full details are given in later sections. The MINI-MANUAL is intended to serve as a quick reference for the knowledgeable user.

In both the MINI-MANUAL and in the detailed sections which follow, a shorthand form is used to indicate the size and order of the array that the code expects. This information is enclosed in square brackets immediately after the array name. Essential features are:

1. A single entry in the brackets is the array length.

2. No brackets at all indicates a simple variable (i.e., an array of unit length).

3. A dash (-) in the brackets indicates an arbitrary length.

4. A semicolon (;) indicates that the input for that array is expected in strings. To the left of the semicolon is the string length. To the right of the semicolon is the number of strings in the array.

5. If the number of strings is shown as a product, the order is important. The leftmost quantity must be exhausted first, then, the next one to the right is varied. For example, the array name for the full spatial source distribution is shown as:

   SOURCF [IT;JT*NMQ]

   where - IT is the number of fine meshes in the X-direction, JT is the number of fine meshes in the Y-direction, and NMQ is the number of input source moments. For this array, the first string is composed of the P_0 source values for each x mesh point in the first y mesh. The next string is the P_0 source values in the second y mesh. This process is repeated for all JT y meshes. Then starting again with the first y mesh, the P_1 source values for each x mesh are given. After all P_1 values are given, the P_2 values follow. Continue until all NMQ moments are specified.

Note: Usually, values for the quantities within brackets will have already been specified in the input. Sometimes, however, a quantity is derived from the array input itself. For instance, in this particular case, NMQ is not an input quantity; rather, the code counts the number of strings and then, knowing JT, deduces what NMQ must have been.

MINI MANUAL

Title Line Control
(3I6 Format)
NHEAD,NOTTY,NOLIST

Title Line(s)

(IF NHEAD>0)

Block-I:Controls & Dimensions

IGEOM
NGROUP
ISN
NISO
MT
NZONE
IM
IT
JM
JT

MAXLCM
MAXSCM

-- -- -- -- --

NOSOLV
NOEDIT

NOGEOD
NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM

T

Block-II:Geometry

XYEPS

XVERT [-;-]
YVERT [-;-]
SMNAME [1;-]
SMIT [1;-]
SMINTS [-;-]
SMNCORS [-;-]

SMZONES [-;-]
SMZNAME [1;-]

OBJECTS [3;-]

COMPONS [5;-]

GEOMETRY [5]

BDRYSEG [5;-]

SMFNAME [-;-]
SMFZNAME [-;-]
OBFNAME [-;-]
CMFNAME [-;-]
GEFNAME [-]

T

Block-III: Cross Sections

LIB
valid: *ODNINP*
XSLIB
ISOTXS
GRUPXS
BXSLIB
MACRXS
MACBCD
XSLIBB
(local)*MENDF*
(local)*MENDFG*
alternate XSLIB name

WRITMXS
valid: *MACBCD*
XSLIBB
XSLIBF
XSLIBE

LNG
BALXS
NTICHI
CHIVEC [NGROUP]
LIBNAME
-- -- -- -- -- -- --
Rest of this block is needed only for text libraries.

MAXORD
IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD (default:1)
NAMES [NISO]
EDNAME [IHT-3]
NTPI [NISO]
VEL [NGROUP]
EBOUND [NGROUP+1]

T

iff LIB= ODNINP, insert ASCII text cross sections here

Block-IV: Mixing

MATLS [-;MT]
ASSIGN [-;NZONE]

PREMIX [-;-]

ASGMOD [-;-]
CMOD

MATNAM [MT]
ZONNAM [NZONE]
MATSPEC [-]
valid: *ATFRAC*
WTFRAC
ATDEN

ATWT [-]

T

Block-V: Solver

IEVT
ISCT
ITH
IBL
IBR
IBT
IBB

EPSI
IITL
IITM
OITM
ITLIM

Solver (continued)

--- Output Controls ---

FLUXP
XSECTP
FISSRP
SOURCP
ANGP
BALP
RAFLUX
RMFLUX

--- Miscellaneous ---

TRCOR
valid: *DIAG*
BHS
CESARO
NO
NORM
BHGT

CHI [NGROUP;M]

DEN [IT;JT]
-or-
DENX [IT], DENY[JT]

Solver (continued)

--- Quadrature -------

GRPSN [NGROUP]
IQUAD
WGT [MM]
MU [MM]
ETA [MM]

--- Flux Guess -------

INFLUX

--- Searches ---------

IPVT
PV
EV
EVM
XLAL
XLAH
XLAX
POD

XM [IM], YM [JM]

Solver (continued)

----Volumetric Source----

INSORS

SOURCE [NGROUP;NMQ]
-or-
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ]
-or-
SOURCE [NGROUP;NMQ] and
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ]
-or-
SOURCF [IT;JT*NGROUP*NMQ]
-or-
SOURCE [NGROUP;NMQ] and
SOURCF [IT;JT*NMQ]

-----Boundary Source-----

SILEFT [NGROUP;JT]
SIRITE [NGROUP;JT]
SIBOTT [NGROUP;IT]
SITOP [NGROUP;IT]
-or-

SALEFT [MM*2;NGROUP*JT]
SARITE [MM*2;NGROUP*JT]
SABOTT [MM*2;NGROUP*IT]
SATOP [MM*2;NGROUP*IT]
-or-

BSLFTG [NGROUP]
BSLFTY [JT]
BSLFTA [MM*2]

BSRITG [NGROUP]
BSRITY [JT]
BSRITA [MM*2]

BSBOTG [NGROUP]
BSBOTX [IT]
BSBOTA [MM*2]

BSTOPG [NGROUP]
BSTOPX [IT]
BSTOPA [MM*2] T

Block-VI: EDIT

PTED
ZNED

POINTS [K], K≤IT*JT
EDZONE [IT;JT]

EDXS [K], K≤NEDT
RESDNT
EDISOS [K], K≤NISO
EDCONS [K], K≤NISO
EDMATS [K], K≤MT
XDF [IT]
YDF [JT]

RSFE [NGROUP;-]
RSFX [IT;-]
RSFY [JT;-]
RSFNAM [-]

ICOLL [K], K≤NGROUP
IGRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

PRPLTED

T

TWODANT/GQ INPUT DETAILS

Introduction

The following pages of this section give details for each of the input arrays. All valid TWODANT/GQ arrays are discussed in this section in detail complete enough to form the input.

However, the beginning user, particularly one unfamiliar with discrete-ordinates codes, may find that he is missing some information of a background nature. See "What Is Available Elsewhere" on page 4-14 for that.

First, here are a few general instructions:

1. All six of the input blocks are normally included. Block-I is always required but any of the other five blocks may be omitted under the proper conditions. The input module reads each block in turn and from it generates one or more binary interface files. The interface files drive the SOLVER and EDIT modules. Thus, if the user wants no edits, the Block-VI input may be omitted. Then with no interface file, the EDIT module will not be executed. Alternatively, if the interface file is available from another source, the corresponding block of input may be omitted. For instance, Block-II describes the geometry. The input module normally writes this information to the GEODST interface file. If the GEODST file is available from another source or a previous run, the Block-II input may be omitted.

2. A general theme of the TWODANT/GQ input is that arrays that are not needed are not entered. Presence of an array indicates that it should be used. Thus, for example, if the density array is entered (DEN array), the cross section at each mesh interval will be modified accordingly. No separate switch need be set to say that the calculation should be done. To eliminate the density modification, simply remove the DEN array from the input or comment it out.

3. The arrays, in general, are grouped in the input instructions according to function. Thus, for example, the input arrays for the volumetric source are found in a single table, or grouping, of input.

4. Groupings of input data may be marked as "Required" or "Optional" in order to guide the user and speed navigation through the input instructions.

   "Required" means that at least one of the arrays in the grouping must be entered. Thus, you must read through the grouping and enter at least one of the arrays found there.

Groupings marked "Optional" may be skipped if the subject is inappropriate. Thus, using the previous example, if one has no volumetric source, one simply skips to the next grouping of input; there is no need to read about any of the arrays within the volumetric source grouping.

Arrays in groupings not marked as "Required" or "Optional" should be reviewed. These groupings contain arrays of vital data that are used in every calculation, but have default values. Thus, although you may not make any input to these arrays and they are in that sense optional, you must concern yourself with them to ensure that the default values are what is intended.

5. Input arrays may also be marked individually. If not marked, they inherit the marking of the grouping in which they are contained. Thus, an unmarked array in a "Required" grouping is required input, and you must enter that array. An unmarked array in an "Optional" grouping is optional.

   You may encounter a "Required" array within an "Optional" grouping. That means that if you decide to invoke the option represented by that grouping, you must input that particular array. For example, if you want user defined response function reaction rates calculated, you must input the RSFE array.

   All arrays within unmarked groupings are optional. However, values in these arrays may be used by the code, so you should concern yourself with the default values if you choose not to enter a value.

6. Unless specifically noted otherwise, the default on all numeric inputs is zero.

7. In an adjoint run, none of the groupwise input arrays should be inverted. The code will externally identify all groups by the physical group number, not by the calculational group number (the calculational group number is in inverse order). Thus, the user interface should be consistently in the physical group order.

8. The use of information within square brackets to indicate the size of arrays and strings and the order within those arrays is the same as described in "MINI-MANUAL Introduction" on page 4-28.

9. Except where noted, arrays and strings must contain the exact number expected by the code (as indicated in the array or string description). If not, the code will eventually abort with a (hopefully) descriptive error message or messages.

10. New users reading these instructions for the first time and unfamiliar with the TWODANT/GQ input may find it helpful to follow the sample input in Appendix A while reading this section.

11. Array names are shown here in upper case. What you should actually input for them will depend upon the code's implementation on your platform. At the present time, on most platforms, you should use lower case input.

12. Items in italics in the input instructions indicate actual values that may be entered for an array. You will frequently find switches where the input is the digit 0 or the digit 1. This will be represented by *0/1* in the input description. In other arrays where an exact character string is required such as "ISOTXS" in the LIB array, you will find the notation *ISOTXS*. Note that in this notation, the word is both upper case and italicized. This combination means you must enter exactly those characters. Again, although the characters will be shown here in upper case, what you should actually input for them will depend upon the code's implementation on your platform.

13. When a template for the input form is given, as for the MATLS array, the style in the template tells the user what is expected. If an input word or value is lower case and italicized, the user is to replace that position with the entry of his choice. If the input word is in italicized style and in upper case, the user is to input exactly those characters to achieve the desired result. Depending on the implementation on your platform, the input word, itself, is usually in lower case.

14. Units to be used for the input quantities are not spelled out as they only need to be self consistent. However, the following are commonly used: Dimensions in centimeters, isotopic cross sections in barns per atom; then it follows that atom densities are in atoms per barn-centimeter. Sources are particles per cm^3 per second for volumetric sources and particles per cm^2 per second for boundary sources; fluxes will then be in particles per cm^2 per second.

Title Line Details

Title Line Control (format 3I6) {Required}

Word	Name	Comments
1	NHEAD	Number of title lines that follow[a]
2	NOTTY	Suppress output to on-line user terminal? 0/1 = no/yes
3	NOLIST	Suppress listing of all ASCII text input? 0/1 = no/yes (default=no)

a. Follow this control line with NHEAD title lines containing descriptive comments. Format is 12A6.

Block-I Details: Dimensions and Controls

Dimensions {Required}

Name	Comments
IGEOM	*Geometry*: 106/107 = generalized x-y/r-z or use one of the following character strings: *X-Y-GQ, R-Z-GQ*
NGROUP	Number of energy groups.
ISN	S_n order to be used. If ISN is negative, code will use the Chebychev-Legendre (IQUAD=2) quadrature set. (See IQUAD input on page 4-62.)
NISO	Number of physical isotopes on the basic input cross-section library.
MT	Number of physical materials[a] to be created.
NZONE	Number of geometric zones[b] in problem.
IT	Number of mesh intervals in the x (or r) direction in the largest submesh.
JT	Number of mesh intervals in the y (or z) direction in the largest submesh.

a. Material is defined on page 4-53.

b. Zone is defined on page 7-13.

Storage Requirements {Optional}

Name	Comments
MAXSCM	Length of SCM desired (default=40000_{10})
MAXLCM	Length of LCM desired (default=140000_{10})

The above input (Dimensions plus Storage Requirements) for Block-I will cause the code to attempt to produce a full run, subject to availability of the input normally found in the other Blocks. The controls below allow shortened print files, partial runs (say, of only the input module), or cause the code to ignore any of the other input Blocks present. For full details on their use, see "PIECEWISE EXECUTION" on page 13-19.

Run Configuration Controls {Optional}

Name	Comments
NOSOLV	Suppress solver module execution. *0/1* = no/yes.
NOEDIT	Suppress edit module execution. *0/1* = no/yes.
NOGEOD	Suppress writing GEODST file even though the geometry input (Block-II) may be present. *0/1* = no/yes.
NOMIX	Suppress writing mixing files even though the mixing input in Block-IV may be present. *0/1* = no/yes.
NOASG	Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. *0/1* = no/yes.
NOMACR	Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. *0/1* = no/yes.
NOSLNP	Suppress writing the SOLINP file even though Block-V may be present. *0/1* = no/yes.
NOEDTT	Suppress writing the EDITIT file even though Block-VI may be present. *0/1* = no/yes.
NOADJM	Suppress writing the ADJMAC file even though an adjoint calculation is called for. *0/1* = no/yes.

Note: Default on all these controls is no.

Block-II Details: Geometry*

This input block describes the geometry of the problem that TWODANT/GQ will solve. The user has two choices. If adequate for the user's purpose, the standard TWODANT geometry input may be used. However, for the more complicated geometries that TWODANT/GQ can use, the input described below must be used.

Spatial Resolution {Required}

Name	Comments
XYEPS	Spatial resolution. If two points are within XYEPS of each other, they are considered to be the same point. If a point lies within XYEPS of a line, it is considered to be on the line.

In the general case, the code will be putting together submeshes to form the complete mesh. One part of this process is to determine if a point in one submesh is the same point as one in another submesh. Because of limits on the precision of arithmetic calculations and on the input precision, two points that were intended to be the same may be slightly different. We thus introduce the idea that if they are close enough, we shall consider them to be the same point. The user must supply the value XYEPS signifying what is "close enough."

* The information entered in this block is written to an extended version of the CCCC standard interface file GEODST. See note on units on page 4-35.

Submesh Definitions {Required}

Name	Comments
XVERT [-;N[a]]	List of the x coordinates of all the vertices in each submesh.
YVERT [-;N]	List of the y coordinates of all the vertices in each submesh. Must be the same number and order as in XVERT.
SMNAME [1;N] {optional}	Name (up to 8 characters) of each submesh. (default names = submsh1, submsh2, etc.)
SMIT [1;N] {optional[b]}	Row length (in number of intervals) for each submesh. Presence of this input indicates order and a rectangular structure in the XVERT and YVERT arrays. The first ($SMIT_i$+1) vertexes are the vertexes along the (logical) bottom of the first row of intervals in the i-th submesh. The second ($SMIT_i$+1) vertexes are those along the bottom of the second row of intervals (and the tops of the first row of intervals). If SMIT is entered, then SMINTS (below) is not entered. If SMIT is not entered, then the submesh domain may be other than logically rectangular and SMINTS (and optionally SMNCORS) must be entered.
SMINTS [-;N] {optional[b]}	Individual mesh interval definitions for the intervals within each submesh. A mesh interval is an area of the problem surrounded by vertexes, with no interior edges. Enter the list of vertex numbers (indexes in the XVERT array) surrounding each interval. Vertexes must be in order along the interval periphery, either clockwise or counterclockwise. Enter the list of vertexes for interval number 2 immediately after those for interval 1. The input order defines the enumeration of the intervals.
SMNCORS [-;N] {optional}	Number of vertexes surrounding each interval within each submesh. Used to extract interval definitions from the packed SMINTS array. (Default is 4 for every interval).

a. N is the number of submeshes. N is determined by the number of strings in this input array and is not directly entered by the user. In our notation, this would normally be shown as a hyphen, but we give it a name here to indicate that all the other arrays in this grouping must have the same number of strings.

b. Either SMIT or SMINTS must be in the input.

Zoning Definitions {Required}

Name	Comments
SMZONES [M;NSMZ]	Zone number for each of the M_n intervals in a submesh.[a] NSMZ is the number of zonings described. This array is identical in order to the ZONES array in Block-II of the TWODANT input. However, it is all one string for a submesh and not a string per row. This array defines the mesh intervals to which cross-section materials are assigned. The zone number[b] must not be greater than NZONE.
SMZNAME [1;NSMZ]	Name (up to 8 characters) of each zoning. (default names = zoning1, zoning2, etc.)

a. The number of entries in the string must fit some submesh, i.e., the number must match the number of intervals described in some submesh definition.

b. A zone number of zero indicates the mesh contains a void, and no cross section will be associated with that mesh. The zero zone number is not counted in the total zone count NZONE.

Object Definitions {Required[a]}

Name	Comments
OBJECTS [3;NOBJS]	For each of the NOBJS objects, enter the string: *objectname submeshname zoningname* ; where - *objectname* is the name of the object being defined, *submeshname* is the name of a submesh from one of those in the SMNAME array, and *zoningname* is the name of the zoning from one of those in the SMZNAME array.

a. In the special situation where the only zoning in the input fits the only submesh in the input, a default object with the name "object" will be constructed by the code.

Component Definitions {Optional}

Name	Comments
COMPONS [6[a];NC]	For each of the NC components, enter the following string: *objectname* x y θ x_t y_t ; where - (x,y) are the coordinates of a center of rotation in the object named *objectname*. θ is the number of degrees to rotate the object clockwise. x_t and y_t are distances to translate the rotated object in the x and y directions, respectively. These specifications establish the object in the global coordinate system and define a component. (The default component names are compon1, compon2, etc.) If you wish to name the component explicitly, precede the objectname with the component name as in: *componentname objectname* x y θ x_t y_t ; The preceding has shown how to define a component consisting of a single object. But, in general, a component may contain more than one object and may contain other components, as well. The fully general form of the string defining a component is then of the form: *componentname* $name_1$ x_1 y_1 θ_1 x_{t1} y_{t1} $name_2$ x_2 y_2 θ_2 x_{t2} y_{t2} $name_3$ x_3 y_3 θ_3 x_{t3} y_{t3} ••• ; where - $name_i$ may be either an object name or a component name and ••• signifies possible continuation of the pattern.

a. Strictly speaking, 6 is the proper length for a single object in the component with the component name defaulted. If you are using the second form to name a single object component explicitly, the string length will be 7. If you have n subcomponents in the component as in the third form, then the length will be 6*n or 6*n+1, depending on whether you explicitly name the component.

Geometry Definition {Required[a]}

Name	Comments[b]
GEOMETRY [5]	For the geometry enter the following sequence: *name* x y θ x_t y_t ; where - (x,y) are the coordinates of a center of rotation in the object or component named *name*. θ is the number of degrees to rotate the object/component clockwise. x_t and y_t are distances to translate the rotated object/component in the x and y directions, respectively. These specifications establish the object/component in the global coordinate system. The preceding has shown how to describe a geometry consisting of a single object/component. But, in general, the geometry will contain more than one object/component. The more general form of the sequence defining the geometry is then of the form: $name_1$ x_1 y_1 θ_1 x_{t1} y_{t1} $name_2$ x_2 y_2 θ_2 x_{t2} y_{t2} $name_3$ x_3 y_3 θ_3 x_{t3} y_{t3} ••• ; where - $name_i$ may be either an object name or a component name and ••• signifies possible continuation of the pattern.

a. For the special case where a single submesh and a single zoning are described and the submesh fits the problem domain (i.e., it is IT by JT), only the XVERT, YVERT, and SMZONES arrays are required. No other geometry input is required unless the user chooses to explicitly describe the boundary conditions with the BDRYSEG array.

b. Input for GEOMETRY follows the same format as that for a component description. The geometry is really just the highest level component.

Boundary Conditions {Optional[a]}

Name	Comments
BDRYSEG [5[b];NBD]	For each of the NBD boundary segments[c] enter the following string: x_1 y_1 x_2 y_2 *ib* *[ir]* ; where - (x_1, y_1) are the coordinates of the starting point of the boundary segment, (x_2, y_2) are the coordinates of the ending point of the boundary segment, *ib* is an integer describing the boundary condition applying on this segment and the optional *ir* (used only if ib=7 or ib=8) is the index of the recipient segment for fluxes exiting this segment. *ib* must be one of the values defined in the CCCC GEODST file definition (e.g., 0 = zero flux, 1 = reflection, 2 = no return, or extrapolated zero flux for diffusion) or additionally, 7 or 8. 7 or 8 means flux exiting on this segment enters on segment ir. If *ib*=7, the exiting flux is rotated through the appropriate angle for entry on the recipient segment. This handles translation and rotation (i.e., tiling) as well as the rotationally periodic case. If *ib*=8, the exiting angular flux is first mirrored about the perpendicular to the exit segment before being rotated and entered on the recipient segment. This handles a reflective type of symmetry.

a. If not entered, code will look for IBL, IBR, IBT, and/or IBB in the Block-V (solver) input. If none of them is present, the default will be vacuum all around.

b. The string length will be 6 if ib=7 or ib=8.

c. Boundary segments have sense. In traveling from point 1 to point 2 along a segment, the interior will lie to the left and the exterior will lie to the right.

Long Name Aliases {Optional}

Name	Comments
SMFNAME	Long name for submeshes
SMFZNAME	Long name for submesh zonings
OBFNAME	Long name for objects
CMFNAME	Long name for components
GEFNAME	Long name for the geometry

A long name alias is a string of characters not restricted to the usual 8 characters that can alternately be used as the name of an entity. These names are carried along for future descriptive purposes only. Where names are specified elsewhere in the input, they must be the short names. As with short names, names with embedded blanks may be used if they are double quoted.

Block-III Details: Nuclear Data

Nuclear Data Type and Options {Required}

Name	Comments
LIB	Name[a] and form of the cross-section data file. Enter as a data item one of the following words:
	Word — Description
	ISOTXS[b] — CCCC standard isotope ordered binary cross-section file.
	XSLIB — ASCII text library supplied in a separate file named XSLIB.
	ODNINP — ASCII text library follows after this block of input (after the T of Block-III).
	GRUPXS[c] — CCCC standard group ordered cross-section file.
	BXSLIB — Binary library supplied as a separate file named BXSLIB. [See "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 10-12.
	MACRXS[d] — Use existing files named MACRXS for SOLVER module, SNXEDT for EDIT module. These files were created in a previous run. 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.
	XSLIBB — See "XSLIBB Card-Image Library File" on page 10-12.
	MACBCD — ASCII form of MACRXS file.
	MENDF — (LANL only) See "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.
	MENDFG — (LANL only) See "The Los Alamos MENDF5G Gamma Cross-Section Library" on page 10-14.
	other — If a word other than those listed above is entered, the code will use the file with that word as its name, provided that file exists in the user's file space. Such a file must be structured as an XSLIB file.
WRITMXS {optional}	Controls the code's writing certain ASCII cross-section files.[e] Enter one of the following words:

Nuclear Data Type and Options (Cont.) {Required}

Name	Comments
	Word — Description
	MACBCD — Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file.
	XSLIBB — Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file.
	XSLIBE — Creates the cross-section file named XSLIBE, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBE is in Los Alamos 6E12 format (IFIDO=0).
	XSLIBF — Creates the cross-section file named XSLIBF, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBF is in FIDO fixed-field format (IFIDO=1).
LNG {optional}	Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.
BALXS {optional}	cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using! Value — Description *-1* — balance cross sections by adjusting absorption cross section. *0* — do not balance cross sections. (default) *1* — balance cross sections by adjusting self-scattering cross section.
NTICHI {optional}	MENDF fission fraction to be used for the problem (LANL only). *1/2/3* = Pu239/U235/U238 (default is U235). Will be overridden by any CHIVEC input described below or by any zone-dependent CHI in input Block-V.
CHIVEC [NGROUP] {optional}	Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.

a. On UNIX systems, the user may specify a search path for some of these files using the environment variable SNXSPATH. See "Library Search Path" on page 4-96 for details.

b. The CCCC standard for file ISOTXS does not allow the inclusion of the 2L+1 term in the higher order scattering cross section. However, if you have a nonstandard file which contains the 2L+1 term, you may override by setting I2LP1=1. See "Text Cross-Section Library Format" on page 4-50. TWODANT/GQ will then convert the cross sections to the appropriate internal form.

c. The 2L+1 term on GRUPXS is treated the same as for ISOTXS. See footnote b.

d. In the convention used in this user's guide, a MACRXS library contains "material" cross sections; all the other libraries contain "isotope" cross sections.

e. See "COUPLED NEUTRON-GAMMA CROSS SECTIONS" on page 10-15.

Alternate Library Name {Optional}

Name	Comments
LIBNAME	Alternate name of the library file. May be used only with certain types of libraries. See Table 4.1.

The entries in the LIB input variable normally dictate both the form and the name of the cross-section library. If the user specified ISOTXS, for example, the code would look for a file named ISOTXS and expect it to be in the CCCC format for an ISOTXS file.

For some libraries, the user may specify the form in the LIB array and specify separately the name in the LIBNAME array. The libraries that can be treated this way are shown in Table 4.1.

Table 4.1 LIBNAME Availability

LIB	LIBNAME AVAILABLE?
MACRXS	No
GRUPXS	Yes
ISOTXS	Yes
BXSLIB	Yes
ODNINP	No
MACBCD	No
XSLIBB	No
MENDF[a]	No
MENDFG[b]	No
XSLIB	Yes
other	Ignored

a. Available only at Los Alamos.

b. Available only at Los Alamos.

The BXSLIB file requires special treatment. It is normally created when the original library is a text library in the input stream or in the XSLIB form. In subsequent runs, this binary BXSLIB file may be used as the source of the cross-section data. The user may wish to save this file under another name. The program, in future runs, may then access the library for reading by using LIBNAME to specify that name.

This procedure is wise because some cases using the BXSLIB form as input also require rewriting it in order to add new information. When this situation arises, the rewritten file is always named BXSLIB. Thus, if the original BXSLIB form library had a different name, it would be protected from being overwritten. For the remainder of the current run, the program will access the file named BXSLIB.

Text Cross-Section Library Format {Required if LIB= XSLIB or LIB=ODNINP}

Name	Comments
MAXORD	Highest Legendre order in the scattering tables.
IHM	Number of positions (entries) in each row of the cross-section table.
IHT	Position number of the total cross section.
IHS {optional}	Position number of the self-scatter cross section. (default = IHT + 1).
IFIDO {optional}	Format of the cross-section library. *-1/0/1/2* = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.
ITITL {optional}	A title line precedes each table. *0/1* = no/yes
I2LP1 {optional}	Higher order scattering cross sections on the library contain the 2L+1 term. *0/1* = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.
SAVBXS {optional}	Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. *0/1* = no/yes.
KWIKRD {optional}	Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? *0/1* = no/yes (default=yes).
NAMES [NISO] {optional}	Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).
EDNAME [IHT-3] {optional}	Character name for each of the EDIT cross-section positions used in the cross-section edits. These are the positions before the absorption cross section in the cross-section table. (default names are: EDIT1, EDIT2, . . .etc.).
NTPI [NISO] {optional}	Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).
VEL [NGROUP] {optional}	Speeds for each group. Needed only for alpha calculations.
EBOUND [NGROUP+1] {optional}	Energy boundaries for each group.

ASCII text libraries may be entered in one of the four forms indicated by the IFIDO input. All four forms share the following features: Cross sections are entered in a table optionally preceded by a title line. A table consists of NGROUP rows of entries. Each row contains the cross sections for a single group and consists of IHM entries. The user specifies the positions in the row occupied by the total and selfscattering cross sections. Order within a row (e.g., for group g) is then as follows:

$$\ldots \sigma_{abs},\ \nu\sigma_{f},\ \sigma_{total},\ \ldots \sigma_{g+2\rightarrow g},\ \sigma_{g+1\rightarrow g},\ \sigma_{g\rightarrow g},\ \sigma_{g-1\rightarrow g},\ \sigma_{g-2\rightarrow g},\ \text{etc.}$$

Notice that all terms in the scattering matrix are in positions relative to that of the self-scattering position and the rest of the cross sections are in positions relative to the position of the total cross section. The positions before the absorption cross section are frequently used for edit cross sections. For more detail, see "Ordering of Cross Sections Within a Cross-Section Table" on page 10-10.

Different Legendre orders are in different tables, which follow in order.

The user may order the group structure either by increasing energy or by decreasing energy. However, it is conventional and desirable for most problems to order it by decreasing energy, that is, group one is the highest energy. In that case, the scattering cross sections to the left of $\sigma_{g\rightarrow g}$ such as $\sigma_{g+1\rightarrow g}$ are upscattering terms and the terms to the right of $\sigma_{g\rightarrow g}$ are the downscattering terms.

In the Los Alamos format, the table is entered with a standard Fortran 6E12 format.

For greater precision in your input, use the 4E18 option.

In the fixed field FIDO format that ANISN[7] uses, entries are made in six twelve-column fields. Each twelve-column field is divided into three subfields, a two-column numeric field, a one-column character field, and a nine-column numeric field. See page 9-19 for details if you are not familiar with this input. The last field in each table must have the character T in the character position. No array identifier should be used. This format also restricts the usable input operators to T, *, R, -, +, and Z.

In the free field form, entries do not have to be in designated columns. Rather, the rules specified in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 apply. Each table in this form is also terminated with the character T. No array identifier (i.e., array name with appended equals sign) should be used.

Block-IV Details: Cross-Section Mixing

A short summary of the primary mixing arrays, MATLS and ASSIGN, is given here for quick reference. Normally, THESE TWO ARRAYS ARE REQUIRED and, in most problems, would be the only arrays in this block. Other mixing arrays are also briefly described.

There are actually several nested levels of mixing. Each level has the job of calculating values from expressions of the form: $\Sigma_g = \sum_{i=1}^{k} N_i \sigma_{i,g}$ for each group, g. The user's job is to input the N_i for all the k components of the mixture and to specify each component, i. Component i has the cross section, $\sigma_{i,g}$. In common usage, for the first level of mixing, $\sigma_{i,g}$ is the effective microscopic cross section and N_i is the atom density of isotope i, and Σ_g is then the macroscopic cross section of some material. In a higher level of mixing, these materials may be homogenized into a single material by using their volume fractions for the N_i. With several nested levels, the user has a great deal of flexibility in defining what Σ_g is for that level. A more complete discussion of mixing will be found in the chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1.

A discussion of cross section processing is outside the scope of this document, but it should be noted that the user needs to be aware of the processing that is inherent in the input library. For instance, for materials in which there are isotopes with cross-section resonances, self shielding of the cross sections for these isotopes may be important and this effect must have been considered in the preparation of the "effective" microscopic cross sections for these isotopes. Since the self shielding is dependent on the amounts and types of the other isotopes in the material, the "effective" cross section is strictly valid only for use in a mixture which has the same composition as was used in the self shielding calculation. If the user desires to use this same "effective" microscopic cross section in some other composition (mix) of material, it is up to the user to verify the accuracy of this approach.

Primary Mixing Arrays {Required}

Name	Description
MATLS[a] [-;MT]	Instructions for mixing "isotopes" or premixes into "materials." See details below.
ASSIGN[b] [-;NZONE]	Assignments of materials to geometric zones. See below.
PREMIX [-;-] {optional}	Instructions for mixing "isotopes" into premixes. See below.

a. The information entered in the MATLS= array is written to the CCCC standard interface files NDX-SRF and ZNATDN.

b. Information entered in the ASSIGN= array is written to the code-dependent interface file ASGMAT.

In order to understand how cross sections are mixed and the resultant material placed in the problem, we first need a little conceptual information.

The key entities used in specifying the cross-section spatial distribution are coarse mesh, zone, isotope, and material.

The basic geometry of the problem is defined with the coarse meshes specified in Block-II. The geometric areas called zones are also defined there using the ZONES array; the ZONES array designates the zone number assigned to each coarse mesh.

Here in Block-IV, we mix cross sections and assign them to the zones created in Block-II. For the purposes of this discussion, the cross sections found on the input library belong, by definition, to "isotopes," no matter what their true nature. These "isotopes" may then be mixed to form materials, using the MATLS array. Materials are then assigned to zones using the ASSIGN array.

MATLS input array

The general form of a MATLS mix instruction is shown below:

$$\text{MATLS=}\ \ mat_1\ \ comp_1\ den_1,\ \ comp_2\ den_2,\ \text{...etc....}\ ;$$

where mat_1 is the desired character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional materials (i.e., mat_2, mat_3, and so on up to the required number, MT) are defined in subsequent strings. Each string may contain as many components as necessary (actual limit = 500). A component is usually an isotope from the library, but may also be a temporary material created by the PREMIX array (see below).

When the component is an isotope, the den_i is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 4-58).

Short form: MATLS= *ISOS*

This form specifies that there should be as many materials as isotopes and that isotope number 1 is to be used for material number 1, isotope number 2 is to be used for material number 2, and so on.

In the special case where there is only a single component in a material and its density is unity, the density entry may be omitted as in the first material below:

MATLS= mat_1 $comp_1$; mat_2 $comp_2$ den_2; ... etc. ... ;

ASSIGN input array

The general form of the ASSIGN instruction is shown below:

ASSIGN= $zone_1$ mat_1 vol_1, mat_2 vol_2, ... etc. ... ;

where $zone_1$ is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). mat_1, mat_2, and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" vol_1, vol_2, and so on. Additional zones (i.e., $zone_2$, $zone_3$, and so on up to the required number, NZONE) are defined in subsequent strings. Although it is highly recommended that you use character names, here it is convenient to use the numeral for the zone name because it is the same numeral entered in the ZONES array.

Short form: ASSIGN= *MATLS*

This form specifies that there are as many zones as there are materials, and that material number 1 is to be assigned to zone number 1, material number 2 to zone number 2, and so on.

NOTE: The short form ASSIGN=*MATLS* can not be used if you intend to use the ASGMOD input array described later in this section.

PREMIX input array

The PREMIX array forms temporary materials in a way exactly analogous to the way that permanent materials are formed in the MATLS array. The difference in treatment is that the temporary materials created by PREMIX exist only long enough to complete the mixing; they are not available for assignment to geometric zones, nor are they available for use in material edits.

The general form of a PREMIX mix instruction is shown below:

PREMIX= $tmat_1$ $comp_1$ den_1, $comp_2$ den_2, ... etc. ... ;

where $tmat_1$ is the character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional temporary materials (i.e., $tmat_2$, $tmat_3$, and so on) may be defined in subsequent strings. A component may be either an isotope from the library or another temporary material created by PREMIX.

The PREMIX array is useful for organizing the mixing input. For instance, it is frequently useful to mix the cross sections for a molecule of water and then in subsequent mix instructions, to input the molecular density of water as opposed to entering the atom density for both hydrogen and oxygen. Other examples are to form average cross sections for an element composed of many isotopes, or to form full density materials and then in later mix instructions, to put in the volume fraction of the full density material.

Character Names vs. Numeric Names

In the foregoing discussion, isotopes, materials, and zones were identified by their character names. Optionally, they may be referred to by their ordinal number. Thus, 2 for an isotope name would call for the second isotope on the library. However, this practice is NOT recommended.

THE CHARACTER NAME FORM IS HIGHLY RECOMMENDED. It provides the most straightforward, most readable form. If the character name form is used, the naming input arrays in the following table are not needed.

Using the character name form in one array and the numeric name form in another array is particularly discouraged. However, should one wish to use the numeric form in the MATLS and/or ASSIGN arrays, and then subsequently associate character names with the ordinal numbers, one can use the naming arrays in the following table to do so. This situation could arise if, for some reason, one wanted to use material numbers in the MATLS array, but use character material names in the ASSIGN array.

When the library is of the MENDF form, the character names that must be used for the isotope names are discussed in "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.

Mixing Array for a Concentration Search {Optional}

Name	Description
ASGMOD[a] [-;-]	C_1 parameters used in concentration searches. See the discussion below.

a. The information entered in the ASGMOD array is written to the ASGMAT file together with the information from the ASSIGN and CMOD arrays.

ASGMOD input array

The ASGMOD array is used in conjunction with the ASSIGN array when one wishes to vary the composition of a zone or zones in order to achieve a certain value of k-effective or alpha (i.e., in a concentration search). The concentration (or volume fraction) of material x in zone z is given by the following expression:

$$C(z,x) = C_0(z,x) + C_1(z,x)*CMOD$$

where $C_0(z,x)$ is the base concentration of material x in zone z. This is the concentration (or volume fraction) entered in the ASSIGN array for material x. In these arrays, x is not any kind of an index; correspondence is made by name, rather than by position within the array. Thus, for instance, in a problem that had ten materials, we might only assign one of them to a given zone. It would then probably be in the first position in the ASSIGN array string for that zone even though it might have been, say, sixth in the list of all materials.

$C_1(z,x)$ is the corresponding entry in the ASGMOD array for material x in zone z.

CMOD is the search parameter (sometimes called search eigenvalue) that will be varied by TWODANT/GQ in order to achieve the desired k-effective or alpha value. In a search calculation, the initial value for CMOD will be the input value EV.

The general form of the ASGMOD instruction is shown below:

ASGMOD= *zone* mat_m vol_m, mat_n vol_n, ... etc. ... ;

where *zone* is the character name of any zone in the problem, mat_m, mat_n, and so on are the character names of any of the materials that will be present in this zone, and vol_m, vol_n, and so on are the C_1 values for respectively, mat_m, mat_n, and so on. Additional zones may be specified in subsequent strings. All zones do not have to appear in the ASGMOD array nor do all materials within a zone have to appear in the string for that zone.

Concentration Modifier {Optional}

Name	Description
CMOD	Concentration modifier. Input value is not used in a search. See the discussion below.

The concentration modifier, CMOD, is varied by TWODANT/GQ during a search calculation. For any other type of calculation, a value of CMOD may be input and the composition of the zones will be calculated using the expression above for C(z,x).

Miscellaneous Mixing Input {Optional}

Name	Comments
MATNAM [MT]	Character material names for Materials. Used only if the mat_1 name used in the MATLS array was integer. First entry in MATNAM array is the desired character name for Material number 1, second entry is the desired character name for Material number 2, etc.
ZONNAM [NZONE]	Character zone names for Zones. Used only if the zone name entry in the ASSIGN or ASGMOD array was integer. First entry in the ZONNAM array is the desired character name for Zone number 1, second entry is the desired character name for Zone number 2, etc.
MATSPEC [≤MT]	Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions. Allowable entries are the words: *ATDENS* (default) atomic densities *ATFRAC*[a] atomic fractions *WTFRAC* weight fractions Can be input as a vector with up to MT entries (one for each Material) [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13.] If less than MT entries are made, the last entry will be used to fill out the array to a length of MT.
ATWT [≤2*NISO] {required[b]}	Atomic weights of the isotopes. If using MATSPEC=ATFRAC or WTFRAC, atomic weights must be available to the code. Entries for the ATWT array are made in pairs, as follows: ATWT= $iso_1\ atwt_1\ iso_2\ atwt_2$ .... where iso_n is the isotope name (identifier) for isotope n on the cross-section library and $atwt_n$ is that isotope's atomic weight. [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13].

a. ATFRAC and WTFRAC cannot be used with PREMIX.

b. Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.

Block-V Details: Solver Input

Desired Calculation {All Optional}

Name	Comments
IEVT	Calculation type: Enter one of the following values:. Value — Calculation Desired 0 — inhomogeneous source (default) 1 — k_{eff} 2 — α (time absorption) search 3 — concentration search 4 — dimension search
ISCT	Legendre order of scattering (default=0).
ITH	0/1 = direct/adjoint calculation (default=0).
IBL[a]	Left boundary condition. 0/1/3 = vacuum/reflective/white.
IBR	Right boundary condition. 0/1/3 = vacuum/reflective/white.
IBT	Top boundary condition. 0/1/2/3 = vacuum/reflective/periodic/white.
IBB	Bottom boundary condition. 0/1/2/3 = vacuum/reflective/periodic/white.

a. These are the boundary conditions on the side of the logical mesh, not the physical side. To use a boundary condition on a physical boundary, see "Boundary Conditions" on page 4-44.

IBL, IBR, IBT, and IBB are to be used only when the geometry input is provided via a standard GEODST file, such as one generated by the TWODANT code. In TWODANT/GQ, the geometry is normally provided in Block-II where the boundary conditions are given by the BDRYSEG array.

Iteration Controls
{All Optional}

Name	Comments
EPSI	Convergence precision (default=0.0001).
IITL	Maximum number of inner iterations per group at first (default=1).
IITM	Maximum number of inners allowed when near fission source convergence (default chosen by code).
OITM	Maximum no. of outer iterations (default=20).

Output Controls
{All Optional}

Name	Comments
FLUXP	Final flux print. *0/1/2* = no/isotropic/all moments.
XSECTP	Cross-section print. *0/1/2* = no/principal/all .
FISSRP	Fission source rate print. *0/1* = no/yes.
SOURCP	Source print. *0/1/2/3* = no/as input/normalized/both .
ANGP	Print the angular flux. *0/1* = no/yes. CAUTION! this is very LARGE output. ANGP=1 will also cause the RAFLXM or AAFLXM file to be written.
RAFLUX	Prepare angular flux file (RAFLUX or AAFLUX). *0/1* = no/yes.
RMFLUX	Prepare flux moments file (RMFLUX or AMFLUX). *0/1* = no/yes.

Miscellaneous Solver Input {Optional}

Name	Comments
TRCOR	Apply transport correction[a] to cross sections on MACRXS file. Enter one of the following words: Word — Description *DIA* — Use diagonal transport correction *BHS* — Use Bell-Hansen-Sandmeier correction *CESARO* — Use Cesaro "correction" *NO* — (or omit entry) don't apply correction
NORM	Normalize the fission source rate to this value when IEVT≥1 or normalize the inhomogeneous source rate to this value when IEVT<1. NORM=0 means no normalization. (Integral of source rate over all angle, space, and energy = NORM, except for k_{eff} problems where the integral is equal to NORM*k_{eff}.) Any fluxes printed here (i.e., caused by setting FLUXP nonzero) will be normalized consistently with this source rate.
BHGT	Buckling height to use to correct for leakage. Units are centimeters if macroscopic cross section is in cm^{-1}. Ignored if r-z.
CHI [NGROUP;M]	Fission fraction born into each group[b]. Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.
DEN [IT;JT] or	Density factor to use at each fine mesh point.
DENX [IT][c] and/or	Density factor to use at each fine x-mesh (or r-mesh) (default=1).
DENY [JT]	Density factor to use at each fine y-mesh (or z-mesh) (default=1).

a. For more information, see "Transport Corrections for the Cross Sections (TRCOR)" on page 7-31.

b. This input will override any previous CHI from earlier blocks or from any cross-section library which contains CHI.

c. In this second form, the density factor DEN(i,j,k), at mesh interval (i,j,k) is computed as follows:

$$DEN(i,j) = DENX(i)*DENY(j)$$

Quadrature Details

Name	Description
GRPSN [NGROUP]	S_n order to be used for each group.
IQUAD	Source of quadrature constants. Enter one of the following: Value — Description *-3* — Get constants from SNCONS file. *-2* — Triangular Chebychev-Legendre built-in set. Any even value for ISN can be used up to 50. From 50 to 100, ISN must be in multiples of 10. See Ref. 6 for details. *1* — Traditional built-in constants. Any even value for ISN can be used between 2 and 16, inclusive. (This is the default). *2* — Rectangular Chebychev-Legendre built-in set. (REQUIRES ISN negative in Block-I!) Any even value for the absolute value of ISN can be used up to 50. From 50 to 100, the absolute value of ISN must be in multiples of 10. See Ref. 6 for details.
WGT[a] [MM[b]] {optional}	Quadrature weights.
MU [MM] {optional}	Mu cosines.
ETA [MM] {optional}	Eta cosines.

a. Presence of the WGT, MU, and ETA arrays overrides the IQUAD input.

b. MM=ISN*(ISN+2)/8 for iquad = -2,1.
MM = (ISN/2)**2 for iquad = 2.

Flux Start {Optional}

Name	Comments
INFLUX	0/1 = no/yes Read flux start from the RTFLUX file.[a]

a. There is presently no ASCII text input flux guess available for TWODANT/GQ.

General Eigenvalue Search Control[a] { IEVT >1}

Name	Comments
IPVT	Type of eigenvalue to search for in a concentration or dimension search. *0/1/2* = none / k_{eff} / α. (default = 1).
PV	Value of k_{eff} or α to which to search. (default = 1.0 if IPVT=1, 0.0 if IPVT=2.
EV	Initial search parameter. Value at which to start the search parameter. (default=0).
EVM	Initial search parameter increment. Amount by which to change search parameter in the first step of a search. (REQUIRED - there is no default).
XLAL	Lambda lower limit for search. (default = 0.01).
XLAH	Lambda upper limit for search. (default = 0.5).
XLAX	Lambda convergence criterion for second and subsequent search steps. (default = 10*EPSI).
POD	Parameter oscillation damper. (default=1.0).

a. See "Eigenvalue Searches" on page 7-33 for definitions of these quantities.

TWODANT/GQ can vary the composition of a zone (or zones) or the coarse mesh boundaries in order to achieve a desired k_{eff} or α value. The search input consists of some general search input plus input specific to the type of search being performed.

Dimension Search Input {Required if IEVT=4}

Name	Comments
XM [IM]	x-dimension fractional change per mesh interval (or r).
YM [JM]	y-dimension fractional change per mesh interval (or z).

The dimension search requires the XM and/or YM input as well as the general search input above. During the search, TWODANT/GQ varies the search parameter (sometimes called the search eigenvalue) denoted by EV in the following expressions to change the mesh vertex positions on iteration k to those on iteration k+1:

$$XVERT(i+1,j)^{k+1} = XVERT(i,j)^{k} +$$

$$\{XVERT(i+1,j)-XVERT(i,j)\} * [1.0+EV*XM(i)] \quad i = 1, 2, ..., IM, \; j = 1, 2, ..., JM$$

$$YVERT(i+1,j)^{k+1} = YVERT(i,j)^{k} +$$

$$\{YVERT(i+1,j)-YVERT(i,j)\} * [1.0+EV*XM(i)] \quad i = 1, 2, ..., IM, \; j = 1, 2, ..., JM$$

Although they may seem a bit awkward at first, the user will find these expressions to be quite flexible. With proper choice of the XM(i), and YM(j) values, the user can move any or all of the verticies in a row or column, while allowing others to remain stationary. The quantities in { } in the above expressions are always formed from the original input values.

Concentration Search Input {Required if IEVT=3}

Name	Description
	The sole solver input for a concentration search is to set IEVT = 3 (page 4-61) and input the general eigenvalue search controls. But you must also input the ASGMOD[a] and CMOD arrays in Block-IV.

a. A concentration search involves the mixing instructions. A discussion of these two arrays is found in the mixing input description on page 4-55.

Volumetric Source Options {Optional}

Name	Comments
INSORS	Read source from interface file FIXSRC. 0/1 = no/yes
----- For an ASCII text input source, choose one of the following options:	
Option 1:	
SOURCE [NGROUP; NMQ]	Source spectrum for each of NMQ[a] moments. (Spatial distribution is assumed to be flat with value unity)
Option 2:	(input both arrays)
SOURCX [IT;NMQ][b]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z) spatial distribution for each moment.
(Spectrum is assumed to be flat with value unity)	
Option 3:	(input all three arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCX [IT;NMQ]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z) spatial distribution for each moment.
Option 4:	
SOURCF [IT;JT*NGROUP*NMQ]	Spatial distribution for each row, group, and moment.
Option 5:	(input both arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCF [IT; JT*NMQ]	Spatial distribution for each row and moment.

a. NMQ is not an input value but is computed from the number of strings read. NMQ must correspond exactly to the number of moments in a P_n expansion of the source; e.g. NMQ must equal 1 for P_0, 4 for P_1, 9 for P_2, etc.

b. Only in option 4 is the complete pointwise source array, SOURCF(i,j,g,m), given. In all other cases, it must be formed from the lower dimension arrays that are input. That calculation is done by forming the product of those arrays. Thus, in option 3, where the source spectrum, SOURCE(g,m), and the spatial distributions SOURCX(i,m), SOURCY(j,m), are given (for moment m), the full source at mesh point (i,j) in group g for moment m is calculated as follows:

SOURCF(i,j,g,m) = SOURCE(g,m)*SOURCX(i,m)*SOURCY(j,m)

Boundary Source Input {Optional}

Name	Comments
----- For a text-input source, choose one of the following options:	
Option 1: Isotropic Boundary Source.	
SILEFT [NGROUP;JT]	Isotropic source on the left side. (Spectrum at each y mesh interval) (or z).
SIRITE [NGROUP;JT]	Isotropic source on the right side.
SIBOTT [NGROUP;IT]	Isotropic source on the bottom side.
SITOP [NGROUP;IT]	Isotropic source on the top side.
Option 2: Full Angular Boundary Source.[a]	
SALEFT [MM*2[b];NGROUP*JT]	Angular flux on the left for each angle, group, and y (or z) mesh interval.
SARITE [MM*2;NGROUP*JT]	Angular fluxes on the right side.
SABOTT [MM*2;NGROUP*IT]	Angular fluxes on the bottom side.
SATOP [MM*2;NGROUP*IT]	Angular fluxes on the top side.
Option 3: Boundary Source From Vectors.[a c d]	
BSLFTG [NGROUP]	Spectrum on left side.
BSLFTY [JT]	Spatial distribution on left side.
BSLFTA [MM*2]	Angular distribution on left side.
BSRITG [NGROUP]	Spectrum on right side.
BSRITY [JT]	Spatial distribution on right side.
BSRITA [MM*2]	Angular distribution on right side.
BSBOTG [NGROUP]	Spectrum on bottom side.
BSBOTY [IT]	Spatial distribution on bottom side.
BSBOTA [MM*2]	Angular distribution on bottom side.
BSTOPG [NGROUP]	Spectrum on top side.
BSTOPY [IT]	Spatial distribution on top side.
BSTOPA [MM*2]	Angular distribution on top side.

a. This is the left side of the logical mesh, no matter which way(s) it faces.

b. See "Quadrature Details" on page 4-62 for value of MM.

c. The full angular source is constructed from a product of the vectors. For example, the source in angle m on the left side of mesh row j for group g is (the source construction on the other faces is analogous):

$$S(m,g,j) = BSLFTG(g)*BSLFTY(j)*BSLFTA(m)$$

d. Any vector not entered explicitly is defaulted to unity.

Block-VI Details: Edit Input*

Edit Spatial Specifications {Required[a]}

Name	Comments
PTED	Do edits by fine mesh. *0/1* = no/yes.
ZNED	Do edits by zone. *0/1* = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below).
POINTS[≤IT*JT] {optional}	Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points).
EDZONE [IT;JT] {optional}	Edit zone number for each fine mesh interval. USED ONLY IF ZNED=1. (default= SOLVER zones, see SMZONES array, Block-II on page 4-41).

a. Either PTED or ZNED or both must be unity in order to produce reaction rate edits.

* More details for the input for edits are given in chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

Reaction Rates from Cross Sections[a] {Optional[b]}

Name	Comments
EDXS [≤NEDT] {required[c]}	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 the character name of the cross-section type. See the table "Edit Cross-Section Types by Position and Name" on page 4-70 or "MENDF Library Edit Cross Sections" on page 4-77 for the available names. NEDT is the total number of edit cross-section types available from the input cross-section library. (default = all shown in the table). Note: The cross-section types specified in this array apply to any or all of the following edit forms: RESDNT, EDISOS, EDCONS, EDMATS.
RESDNT	Do edits using the resident macroscopic cross section at each point. *0/1* = no/yes.
EDISOS [≤NISO]	Character names of the isotopes to be used in forming Isotopic reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDCONS [≤NISO]	Character names of the isotopes to be used in forming resident Constituent (partial macroscopic) reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDMATS [≤MT]	Character names of materials to be used in forming Material (macroscopic) reaction rates. The ordinal number may alternately be used, but is not recommended. (default = none).
XDF[d] [IT] YDF [JT]	Fine mesh density factors for the x (or r) and y (or z) directions, respectively. The density factor is used to multiply resident Constituent (see EDCONS), macroscopic (see MACRO), and resident macroscopic (see RESDNT) reaction rates only. (default= all values unity).

a. See chapter "RUNNING THE EDIT MODULE" starting on page 8-1 for further discussion.

b. But either something in this grouping or in the "Reaction Rates from User Response Functions" grouping must be input in order to produce reaction rate edits.

c. You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.

d. If density factors were used in SOLVER to modify the cross sections at each mesh interval, the same density factors must be provided here in the XDF and/or YDF arrays as well. The density factor at mesh interval (i,j) is computed as:

XDF(i)*YDF(j)

Edit Cross-Section Types by Position and Name

CROSS-SECTION INPUT VIA ISOTXS or GRUPXS			CROSS-SECTION INPUT VIA ASCII TEXT		
Type	EDIT Position	Name[a]	Type	EDIT Position	Name
chi	1	CHI.....	not used	1	CHI.....
nu-fission	2	NUSIGF..	nu-fission	2	NUSIGF..
total	3	TOTAL...	total	3	TOTAL...
absorption	4	ABS.....	absorption	4	ABS.....
(n,p)	5	N-PROT..	1[b]	5	EDIT1...[c]
(n,d)	6	N-DEUT..	2	6	EDIT2...
(n,t)	7	N-TRIT..	3	7	EDIT3...
(n,alpha)	8	N-ALPH..	.	.	.
(n,2n)	9	N-2N....	.	.	.
(n,gamma)	10	N-GAMM..	.	.	.
fission	11	N-FISS..	N=IHT-3	4+N	EDITN...
transport	12	TRNSPT..			

a. Names are eight characters. A period within a name in this table denotes a blank.

b. Denotes position in the cross-section table. All cross sections in 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 EDNAME array of Block-III.

Reaction Rates from User Response Functions {Optional[a]}

Name	Comments
RSFE [NGROUP;M] {required}	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 with NGROUP entries beginning with group 1.
RSFX [IT;M][b]	Response function x (or r) distribution for M functions.
RSFY [JT;M]	Response function y (or z) distribution for M functions.
{optional}	The above data are entered as M strings of IT or JT entries beginning with mesh point 1. (default=1.0).
RSFNAM [M]	Character names for the user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM).

a. But either something in this grouping or in the "Reaction Rates from Cross Sections" grouping must be input in order to produce reaction rate edits.

b. The M-th response function at space point (i,j) and energy group g is computed as:

RSFX(i,m)*RSFY(j,m)*RSFE(g,m)

Energy Group Collapse Specifications {Optional}

Name	Comments
ICOLL [NBG]	Edit energy group collapsing option: Number of SOLVER energy groups in each EDIT broad group. The NBG entries must sum to NGROUP. (default = 1 energy group per EDIT broad group).
IGRPED	Print option on energy groups. Enter one of the following values:

Value	Description
0	Print energy group totals only
1	Print broad groups only
2	Print broad groups only (same as 1)
3	Print both broad groups and totals

Reaction Rate Summing {Optional}

Name	Comments
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 "Edit Cross-Section Types by Position and Name" on page 4-70). Each of 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. (for more detail, see "Response Function Summing Options" on page 8-13).
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 page 8-13 for more detail.

Mass Inventories {Optional}

Name	Comments
MASSED	Calculate and print mass inventories by zone. *0/1/2/3* = none/ solver zones/edit zones/both (default=1). This option is active only if atomic weights are present. See ATWT on page 4-58.

Power Normalization {Optional}

Name	Comments
POWER {required}	Normalize to POWER megawatts.[a] All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be normalized. Fluxes are normally not printed here in the EDIT module, although they may be extracted by using a unit response function. Any such fluxes will also be normalized to POWER. Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 4-61).
MEVPER {required}	MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power. For the power calculation, TWODANT/GQ needs to know which cross section is the fission cross section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or GRUPXS library that designation is automatically provided (See "Edit Cross-Section Types by Position and Name" on page 4-70). But if one uses an ASCII text library, either ODNINP or XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 4-50).

a. Note that this normalization is meaningless if you are using the results of an adjoint run.

Miscellaneous Edit Items {Optional}

Name	Comments
RZFLUX	Write the CCCC standard zone[a] flux file RZFLUX or AZFLUX. *0/1* = no/yes.
RZMFLX	Write the code-dependent zone[b] flux moments file RZMFLX or AZMFLX. *0/1* = no/yes.
EDOUTF[c]	ASCII output files control. Enter one of the following values: Value — Description *-3* — Write both EDTOGX (without scalar fluxes) and EDTOUT files. *-2* — Write EDTOGX file (without scalar fluxes). *0* — Write neither file. (default). *1* — Write EDTOUT file. *2* — Write EDTOGX file (with scalar fluxes). *3* — Write both EDTOGX (with scalar fluxes) and EDTOUT files.
BYVOLP	Printed point reaction rates will have been multiplied by the mesh volume. *0/1* = no/yes.
AJED[d]	Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. *0/1* = regular/adjoint.
FLUXONE	Flux override. *0/1* = no/yes. Replaces all the input fluxes by unity. Useful for seeing the cross sections used in cross-section edits. WARNING! Meaningful reaction rates cannot be obtained when this switch is on.

a. RZFLUX and AZFLUX are organized by solver zones.

b. RZMFLX and AXMFLX are organized by solver zones.

c. See "ASCII File Output Capabilities (the EDOUTF Parameter)" on page 8-15.

d. See "Adjoint Edits" on page 8-15.

Special Plot Linkage {Optional}

Name	Comments
PRPLTED	Write an ASCII file of the pointwise reaction rates to link to the TECPLOT© plotting package available commercially for a SUN workstation. *0/1/2/3* = print only/nothing/tecplot file/both print and tecplot file.

To exercise this option, the user must have set PTED=1. The code will calculate reaction rates at all the fine mesh intervals and any POINTS input will be ignored.

To link to the TECPLOT© code, the user chooses option 2 or 3. Separate ASCII files called rsp.dat and med.dat will be written for the response function and material edits, respectively. These files are in input form for the TECPLOT© preprocessor.

If option 0 (print only) is chosen, no TECPLOT© files will be written, but the reaction rates will be printed. The format of this printout is organized in a two-dimensional way unlike the normal printout from the EDIT module.

MENDF Library Edit Cross Sections

Reaction Type	Name	Description
χ	CHI	fission spectrum
$\nu\sigma_f$	NUSIGF	effective nu-sigma-fission
σ_t	TOTAL	Total cross section
σ_a	ABS	absorption[a]
(n,n)	MEND1	elastic scattering
(n,n´)	MEND2	inelastic scattering
(n,2n)	MEND3	n,2n scattering
(n,3n)	MEND4	n,3n scattering
(n,γ)	MEND5	gamma production
(n,α)	MEND6	alpha production
(n,p)	MEND7	proton production
(n,f)	MEND8	direct fission
(n,n´)f	MEND9	second-chance fission
(n,2n)f	MEND10	third-chance fission
(n,F)	N-FISS	[(n,F) = (n,f) + (n,n´)f + (n,2n)f]
χ_p	MEND12	prompt fission spectrum (only for fissionable materials)
χ_t	MEND13	total fission spectrum (only for fissionable materials)

a. σ_a for group g is defined as $\sigma_a = \sigma_t - \sum_{g'} \sigma_{g \to g'}$

When using the Los Alamos MENDF5 cross-section library with the codes, there are numerous edit cross sections available for use in the Edit Module. Since these come from the MENDF file, they are called upon with special character names in the Edit Module as part of the EDXS input.

These names are defined in the table above.

REFERENCES

1. 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.

2. 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.

3. 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).

4. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," *Nucl. Sci. Eng.* 64, 344 (1977).

5. R. E. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Multigroup Diffusion Equation," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp. 340-351.

6. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).

7. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693 (March 1967).

APPENDIX A: SAMPLE INPUT

Sample Problem: Supercell k_{eff} Calculation

Our sample case is a hypothetical supercell consisting of a cluster of seven hexagonal assemblies. A two-dimensional, meshed model of the midplane of the cluster is shown in Figure 4.7. There are two types of assemblies, one a fuel assembly with seven hexagonal subassemblies, and the other a control assembly with a cylindrical control rod at the center surrounded by, respectively, a fluid layer that allows rod movement and a tube of cladding. The repeating pattern of the supercell is shown by the letters A and B. These letters indicate that face A of one supercell is in contact with face B of another supercell.

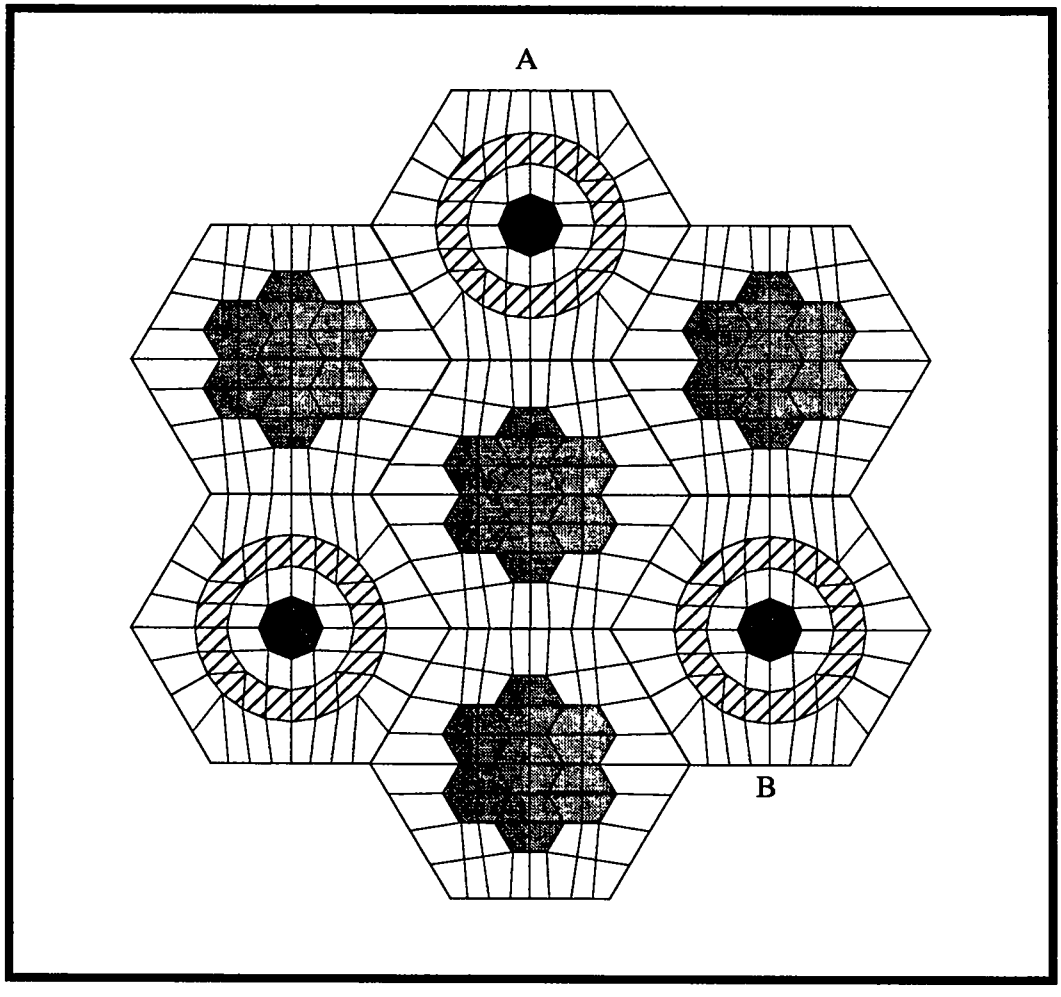


Figure 4.7 Hypothetical Supercell Geometry

Sample Problem: Output Description

Selected items in the output listing are described here. We focus on details of the geometry description. For a more thorough description of output items common to both one and two dimensional calculations, the reader is referred to Appendix A in the chapter "ONEDANT USER'S GUIDE".

The first item provided in the output is a listing of the input lines. This is shown on page 4-85. We see the title lines followed by five Blocks of input. There is no Block-VI input as no edits are wanted. Comments show where Block-VI would be entered.

The Block-I input, commented as such in the input listing, describes controls and dimensions for the problem as a whole. Such items as the geometry option, the angular quadrature, the number of groups, the number of isotopes and materials, and the mesh size are specified here. The mesh size specified here need only be big enough to hold the largest submesh; it does not have to be big enough to hold the final assembled mesh.

The Block-II input describing the geometry follows. We first see the spatial resolution, XYEPS, specified. When mesh points are being matched within the code, two points that are within XYEPS of each other will be considered to be the same point. We then see that the user chosen names of two submeshes are entered and the SMIT input specifies that the two submeshes are both of a rectangular format and have a row length of eight.

The specification of the actual vertex points for both submeshes follows in the XVERT and YVERT arrays. The two submeshes described by these arrays are those for the upper half of the two hexagonal meshes shown in Figure 4.1. We take advantage of the symmetry about the midplane of the objects to be formed in order to economize on the input.

Next follows the user chosen names for the submesh zonings and then, in the SMZONES array, the actual zone numbers for each of the mesh intervals in the submeshes. With the OBJECTS input we then form an object out of each submesh by specifying which zoning to use with that submesh. At this point, we have two objects which are the upper half of the two top objects shown in Figure 4.2.

In the COMPONS array, we assemble two hexagonal components from the objects formed above. Two instantiations of each object are used to form each full component, one of them being first rotated 180 degrees about its origin at the bottom center of the submesh. We now have two components, one the full fuel assembly, and the other, the control assembly.

Then, in the GEOMETRY array, we assemble the complete problem domain, consisting of four instantiations of the fuel component and three instantiations of the control component. The problem domain is now fully described and is as shown in Figure 4.7.

Finally, the boundary conditions are described. We choose to have a boundary segment for each straightline portion of the periphery of the geometry. The endpoints for each of the boundary segments are entered followed by the translational boundary condition denoted by the integer seven, followed by the index of the boundary segment to which it communicates.

After we finish with the geometry description, we describe the cross-section library in Block-III. P_0 cross sections for each isotope are entered in the input stream after Block-III. We give each of the input isotopes a name. The MATLS input then says to use the same names for the names of the materials, that is, do no mixing at this step. These materials are then assigned with appropriate volume fractions to the COOLANT, HOMOFUEL, CONTROL, CTRLTUBE, CTRLCLAD, and CTRLCOOL zones which correspond to the numbers 1 through 6, respectively, in the SMZONES arrays of Block-II.

In Block-V, we specify that this is a k-eigenvalue calculation (IEVT=1). We wish a P_0 calculation of the flux and the fission source is normalized to 1.0.

After the input listing, we start to see the program's interpretation of the input. This starts with the heading "* case title" on page 4-87 and is followed by an interpretation of the Block-I input. Under "* ...block iii - cross section library..." on page 4-87, we have a summary of the cross section library and then under "* ... mixing instructions ..." on page 4-88, we see the mixing input expanded.

The solver input from Block-V is then displayed. Note that under "* ...parameters from block i..." on page 4-90, we see that the number of intervals in each of the x and y directions is shown. This is the size that the program has determined is needed for the assembled complete mesh, and is not the size specified in Block-I, which was only that needed for the largest submesh.

After the material assignment to zones input, together with possible concentration search input is shown on page 4-90, we see a summary of the memory storage requirements and the values used for the quadrature constants.

The material map follows on page 4-91. This shows the material numbers for each of the mesh intervals in the problem domain. Note that the intervals are displayed with the interval count, i, running in the horizontal direction and the interval count, j, running in the vertical direction. Since this may or may not correspond to the actual physical orientations of the mesh lines and intervals, we refer to this sort of a map as the "logical" map of the problem. Note that this particular case does not provide a full, rectangular array of numbers. Rather, there are "logical holes" in the corners.

As far as the rest of the output shown, most of the items are self explanatory and give the same information as a ONEDANT problem except for the two-dimensional differences. However, we would like to focus a little more attention on two items that serve as diagnostics and goodness of solution verification for the problem run: the iteration monitor and the balance table.

In considering the iteration monitor (page 4-92), we recall that for eigenvalue problems we do source iteration for the inner or within-group scattering source and outer iterations for the fission source. We also recall that the default (and recommended) strategy to solve these problems is to do one inner iteration per group (two inners if there are implicit boundary conditions) until the fission source has sufficiently converged, and then to fully converge the inner iterations on the flux. This is reflected in the monitor which is arranged in rows for each outer. The inner convergence is not shown until the fission source has converged to near the input convergence criterion. The first outer, designated outer 1, is a pure diffusion calculation; the rest include the transport sweeps with

DSA acceleration. Each column of the monitor gives respectively, the current CPU time, the transport outer counter, the number of transport inners for this outer (usually equal to the number of groups until the source is converged), the number of multigroup DSA iterations (sub-outers), the eigenvalue estimate at this outer, the precision of the eigenvalue (change from the previous transport outer), the maximum pointwise flux change (not important until the source has converged), the maximum pointwise fission distribution error, and information on the status of the inners. Note that at outer 5 the source has sufficiently converged where now the inner iterations on the groupwise scalar flux can be carried to completion. Thus it is seen that upon completion of outer iteration 6, the whole problem has been converged. From the first column, we see that this was done in 2.06 CPU seconds on the Cray YMP.

The next item is the balance table on page 4-93 which gives quantities from the transport equation integrated over the entire spatial domain of the problem for each group and the sum of the groups. This particle balance is a measure of the integral goodness of the solution and is equal to 1.0-sources/losses. Thus the extraneous source, the fission source, and the inscatter are the source ingredients. The outscatter, net leakage and absorption are the losses. The table for this problem shows that balance is achieved to better than 2 parts in 10^6 for our single group and, since we have only a single group in this testcase, for the total or group summed balance as well. Another thing we would normally notice is the group totals of the inscatter and outscatter. This indicates how well the cross-section set is balanced in the scattering matrix since these two quantities should be the same in the absence of any (n,2n) or (n,3n) reactions. Treatment of the (n,2n) and (n,3n) reactions varies with the cross-section processor, but it is common practice to include the (n,2n) and (n,3n) cross sections in the absorption and total cross sections, thus getting the loss reaction rate correctly and then to include twice the (n,2n) cross section and thrice the (n,3n) cross section in the inscattering cross sections, thus getting the scattering source in the group correctly. In this case, the total inscatter should exceed the total outscatter by the amount of production due to the (n,2n) and (n,3n) reactions. The nprod spectrum gives the number of fission neutrons produced in each group and is useful for determining whether fissions are being produced in the thermal or fast range, for example.

Sample Problem: Output Listing

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*            generalized input module run on 03/17/95 with solver version 01-04-95beta---- release 3.0     machine e
*
*****************************************************************************************************************************
*                  ***************************************************************************************
*                  *                                                                                         *
*                  *                    ...listing of cards in the input stream...                           *
*                  *                                                                                         *
*                  *    1.      2          0                                                                 *
*                  *    2.         twodant/gq test case                                                     *
*                  *    3.  hypothetical seven assembly supercell                                           *
*                  *    4. /                                                                                *
*                  *    5. / the complete geometry is a set of 7 hex assemblies                             *
*                  *    6. / there are two types of assemblies:                                             *
*                  *    7. /   1. control assembly                                                          *
*                  *    8. /   2. fuel assembly with seven hexagonal subassemblies                          *
*                  *    9. /                                                                                *
*                  *   10. / boundary conditions are given explicitly along each face                       *
*                  *   11. / and represent an infinite array of these supercells repeating                  *
*                  *   12. / along a roughly northwest-southeast line                                       *
*                  *   13. /                                                                                *
*                  *   14. / ****** Block-I - Controls and Dimensions ************************              *
*                  *   15. igeom=106 ngroup=1 isn=8  it=8 jt=8 nzone=6 niso=5                               *
*                  *   16.  mt=5 maxscm=89000 maxlcm=30000  t                                               *
*                  *   17. /                                                                                *
*                  *   18. / ****** Block-II - Geometry ***************************************             *
*                  *   19.  xyeps=0.002                          / spatial resolution                       *
*                  *   20.  smname= mshctrl;  mshfuel;           / submesh names                            *
*                  *   21.  smit= 8;  8;                         / row length for each submesh              *
*                  *   22. xvert=                                                                           *
*                  *   23.  /  ** top half of the control assembly **                                       *
*                  *   24.   -4.000  -2.300  -1.400  -0.600   0.000  +0.600  +1.400  +2.300  +4.000        *
*                  *   25.   -3.500  -2.222  -1.293  -0.424   0.000  +0.424  +1.293  +2.222  +3.500        *
*                  *   26.   -3.000  -1.992  -0.990  -0.536   0.000  +0.536  +0.990  +1.992  +3.000        *
*                  *   27.   -2.500  -1.626  -1.150  -0.595   0.000  +0.595  +1.150  +1.626  +2.500        *
*                  *   28.   -2.000  -1.500  -1.000  -0.500   0.000  +0.500  +1.000  +1.500  +2.000;       *
*                  *   29.  /  ** top half of the fuel assembly **                                          *
*                  *   30.   -4.000  -2.000  -1.500  -1.000   0.000  +1.000  +1.500  +2.000  +4.000        *
*                  *   31.   -3.500  -2.500  -1.500  -0.500   0.000  +0.500  +1.500  +2.500  +3.500        *
*                  *   32.   -3.000  -2.000  -1.500  -1.000   0.000  +1.000  +1.500  +2.000  +3.000        *
*                  *   33.   -2.500  -1.750  -1.250  -0.500   0.000  +0.500  +1.250  +1.750  +2.500        *
*                  *   34.   -2.000  -1.500  -1.000  -0.500   0.000  +0.500  +1.000  +1.500  +2.000;       *
*                  *   35. yvert=                                                                           *
*                  *   36.  /  ** top half of the control assembly **                                       *
*                  *   37.   9r0                                                                            *
*                  *   38.   +0.866  +0.595  +0.536  +0.424   0.600  +0.424  +0.536  +0.595  +0.866        *
*                  *   39.   +1.732  +1.150  +0.990  +1.293   1.400  +1.293  +0.990  +1.150  +1.732        *
*                  *   40.   +2.598  +1.626  +1.992  +2.222   2.300  +2.222  +1.992  +1.626  +2.598        *
*                  *   41.   9r3.464;                                                                       *
*                  *   42.  /  ** top half of the fuel assembly **                                          *
*                  *   43.   9r0  9r 0.866 9r 1.732 9r 2.598 9r 3.464;                                      *
*                  *   44. smzname=  zngctrl; zngfuel;           / zoning names                             *
*                  *   45. smzones=                                                                         *
*                  *   46.    6 5 4 3 3 4 5 6   6 5 4 4 4 4 5 6   6 6r5 6   8r6;       / zngctrl          *
*                  *   47.      1 6r2 1   1 6r2 1   3r1 2r2 3r1   8r1;                  / zngfuel          *
*                  *   48. objects=                                                                         *
*                  *   49.    objctrl   mshctrl  zngctrl;                                                  *
*                  *   50.    objfuel   mshfuel  zngfuel;                                                  *
*                  *   51. compons=                                                                         *
*                  *   52.     ctrlhex  objctrl    +0.000 +0.000 +0.000   +0.000 +0.000                    *
*                  *   53.              objctrl    +0.000 +0.000 +180.0   +0.000 +0.000;                   *
*                  *   54.     fuelhex  objfuel    +0.000 +0.000 +0.000   +0.000 +0.000                    *
*                  *   55.              objfuel    +0.000 +0.000 +180.0   +0.000 +0.000;                   *
*                  *   56.                                                                                  *
*                  *   57. geometry=                                                                        *
*                  *   58.               fuelhex     +0.000 +0.000 +0.000   +0.000 +0.000                  *
*                  *   59.               ctrlhex     +0.000 +0.000 +0.000   +0.000 +6.928                  *
*                  *   60.               fuelhex     +0.000 +0.000 +0.000   +6.000 +3.464                  *
*                  *   61.               ctrlhex     +0.000 +0.000 +0.000   +6.000 -3.464                  *
*                  *   62.               fuelhex     +0.000 +0.000 +0.000   +0.000 -6.928                  *
*                  *   63.               ctrlhex     +0.000 +0.000 +0.000   -6.000 -3.464                  *
*                  *   64.               fuelhex     +0.000 +0.000 +0.000   -6.000 +3.464                  *
*                  *   65.  bdryseg=                                 / nww - see orientation              *
*                  *   66.            2.000  10.392   -2.000  10.392  7  2 ;                               *
*                  *   67.            4.000  -6.928    8.000  -6.928  7  1 ;                               *
*                  *   68.           -2.000  10.392   -4.000   6.928  7  4 ;                               *
*                  *   69.            2.000 -10.392   +4.000  -6.928  7  3 ;                               *
*                  *   70.           -4.000  +6.928   -8.000   6.928  7  6 ;                               *
*                  *   71.           -2.000 -10.392   +2.000 -10.392  7  5 ;                               *
*                  *   72. /                                                                                *
*                  *   73.            4.000  +6.928    2.000  10.392  7  8 ;                               *
*                  *   74.          -10.000  -3.464   -8.000  -6.928  7  7 ;                               *
*                  *   75.           -8.000  -6.928   -4.000  -6.928  7 10 ;                               *
*                  *   76.            8.000   6.928    4.000   6.928  7  9 ;                               *
*                  *   77.           -4.000  -6.928   -2.000 -10.392  7 12 ;                               *
*                  *   78.           10.000   3.464   +8.000   6.928  7 11 ;                               *
*                  *   79. /                                                                                *
*                  *   80.           -8.000   6.928  -10.000   3.464  7 14 ;                               *
*                  *   81.            8.000   0.000   10.000   3.464  7 13 ;                               *
*                  *   82.          -10.000   3.464   -8.000   0.000  7 16 ;                               *
*                  *   83.           10.000  -3.464    8.000   0.000  7 15 ;                               *
*                  *   84.           -8.000   0.000  -10.000  -3.464  7 18 ;                               *
*                  *   85.            8.000  -6.928   10.000  -3.464  7 17 ;                               *
*                  *   86.                                                 t                                *
*                  *   87. /                                                                                *
*                  *   88. / ****** Block-III - Cross sections ********************************             *
*                  *   89. lib=odninp,    ihm=6 iht=3 ihs=4 ititl=1,                                        *
*                  *   90.                names= cool fuel clad fluid poison     t                          *
*                  *   91. coolant throughout the supercell                                                 *
*                  *   92.    0.01        0.0         1.0         0.99        0.0         0.0               *
*                  *   93. fuel mixture inside fuel rods                                                    *
```

```
*  94.    0.5        0.57       1.0        0.5        0.0        0.0          *
*  95. cladding material                                                      *
*  96.    0.5        0.0        1.0        0.5        0.0        0.0          *
*  97. fluid surrounding the control rod                                      *
*  98.    0.1        0.0        1.0        0.9        0.0        0.0          *
*  99. poison mixture inside the control rod                                  *
* 100.    0.9        0.0        1.0        0.1        0.0        0.0          *
* 101. /                                                                     *
* 102. / ****** Block-IV - Mixing ***************************************    *
* 103. matls=isos                                    / each isotope is a matl *
* 104. assign=                                       / matls in each zone     *
* 105.        coolant    cool;                                / pure fuel coolant *
* 106.        homofuel   fuel 0.95  cool 0.03  clad 0.02;     / homogenized fuel rod *
* 107.        control    clad 0.05  poison 0.95;              / homogenized control rod *
* 108.        ctrlgap    fluid;                               / control fluid layer *
* 109.        ctrltube   clad;                                / control sheath *
* 110.        ctrlcool   cool;   t                            / control coolant *
* 111. /                                                                     *
* 112. / ****** Block-V - Solution Desired ********************************   *
* 113. ievt=1 isct=0                                 / k-eff, p-zero calculation *
* 114. iitl=1 oitm=100 epsi=1.0e-4 norm=1.0          / convergence controls   *
* 115. chi=1.0                                       / fission fraction       *
* 116.  t                                                                    *
* 117. /                                                                     *
* 118. / ****** Block-VI - No Edits Desired ******************************    *
*                                                                            *
******************************************************************************
```

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                                    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
*
*                          ***************************************************************************
*                          *                                                                         *
*                          *        twodant/gq test case                                             *
*                          *  hypothetical seven assembly supercell                                  *
*                          *                                                                         *
*                          ***************************************************************************
************************
*key end   block i read*
************************
*****************************************************************************************************************************

*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                          ...block i - controls and dimensions...
*
*****************************************************************************************************************************
*
*
*                                              ...dimensions ...
*
*                         106  igeom   6/7/9/11/106/107  x-y/r-z/triangles/r-theta/gq x-y/gq r-z
*                           1  ngroup  number of energy groups
*                           8  isn     angular quadrature order
*                           5  niso    number of input isotopes (from isotxs, grupxs, or cards)
*                           5  mt      number of permanent materials
*                           6  nzone   number of zones
*                           8  im      number of coarse mesh x intervals
*                           8  it      number of fine mesh intervals
*                           8  jm      number of coarse mesh y intervals
*                           8  jt      number of fine mesh y intervals
*
*                                              ...storage...
*
*                                              maxlcm=   30000
*                                              maxscm=   89000
*
*****************************************************************************************************************************
*****************************
*key end   block ii read-geom*
*****************************
*
*                        106  igeom    1/2/3/6/7/8/9/11/14/106/107
*****************************
*key end   block iii read-xs *
*****************************

*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                        ...block iii - cross section library...
*
*****************************************************************************************************************************
*
*                                              ...library source...
*                                                  lib=cdnimp
*
*                                        ...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
*                         3  iht     position of total cross section
*                         4  ihs     position of self scatter cross section
*                         0  ifido   -1/0/1/2 - dtf(4e18.0)/dtf/fixed fido/free fido library
*                         1  ititl   0/1 - no/yes there is a title card before each table
*                         0  i2lp1   0/1 - no/yes library higher order scattering contains 2l+1 factor
*                         0  savxs   0/1 - no/yes save binary xslib file (filename=xslib)
*                         1  kwikrd  0/1 - full fido read/quick fido read (default=quick)
*
*
*                                              ...energy structure...
*
*   group      chi          vel        lower bound   upper bound   group      chi          vel        lower bound   upper bound
*   -----  -----------  -----------  -----------  -----------  -----  -----------  -----------  -----------  -----------
*       1  0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00
*
*                                        last neutron group(lng) is number   1
*
*
*                         0  balxs   -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance
*
*
*                                          ...edit position names...
*
*                                                                 card
*                                        position  edname    position
*                                        --------  -------  --------
*                                               1  chi
*                                               2  nusigf          2
*                                               3  total           3
*                                               4  abs             1
```

```
******************************
*key start card libe read    *
******************************
*
*                                   ...header cards from the card library...
*
*                           isotope isotope
*                           number  name   order        header card
*                           ------  -----  -----  ---------------------------------
*                               1.  cool    p0    -  coolant throughout the supercell
*                               2.  fuel    p0    -  fuel mixture inside fuel rods
*                               3.  clad    p0    -  cladding material
*                               4.  fluid   p0    -  fluid surrounding the control rod
*                               5.  poison  p0    -  poison mixture inside the control rod
******************************
*key end   card libe read    *
******************************
******************************
*key end   block iv read-mats*
******************************
*
*
*
*********************************************************************************************************************************************************
*********************************************************************************************************************************************************


*********************************************************************************************************************************************************
*********************************************************************************************************************************************************
*
*                                          ... mixing instructions ...
*********************************************************************************************************************************************************
*
*          mix      comp       density    comp       density etc.
*          ---      ----       -------    ----       -------
*
*     matls
*     -----
*
*       1. cool     cool     1.00000E+00,
*       2. fuel     fuel     1.00000E+00,
*       3. clad     clad     1.00000E+00,
*       4. fluid    fluid    1.00000E+00,
*       5. poison   poison   1.00000E+00,
*
*     assign
*     ------
*
*       1. coolant  cool     1.00000E+00,       -
*       2. homofuel fuel     9.50000E-01, cool  -  3.00000E-02, clad     2.00000E-02,
*       3. control  clad     5.00000E-02, poison   9.50000E-01,
*       4. ctrlgap  fluid    1.00000E+00,
*       5. ctrltube clad     1.00000E+00,
*       6. ctrlcool cool     1.00000E+00,
************************
*key start mix card xs *
************************
************************
*key end   mix card xs *
************************
*********************************************************************************************************************************************************
******************************
*key end   block v read-solvr*
******************************
************************
*key end   input module*
************************
*
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*                  this twodant/gq problem run on 03/17/95 with solver version 02-16-95beta---- release 3.0      machine e
*       twodant/gq test case
* hypothetical seven assembly supercell
*
*****************************************************************************************************************************************
*
*                                        ...block v -- solver input...
*
*****************************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*                                        ...required input(array name = solin)...
*
*          1        1       ievt   0/1/2/3/4 - type of calculation
*                                      0 inhomogeneous source
*                                      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        0       ibl    0/1/3 - left boundary condition
*                                     vacuum/reflective/white
*          0        0       ibr    0/1/3 - right boundary condition
*                                     vacuum/reflective/white
*          0        0       ibt    0/1/2/3 - top boundary condition
*                                     vacuum/reflective/periodic/white
*          0        0       ibb    0/1/2/3 - bottom boundary condition
*                                     vacuum/reflective/periodic/white
*
*
*                                        ...convergence controls(array name = iter)...
*
*      1.000E-04 1.000E-04  epsi   inner iteration convergence criterion (default=0.0001)
*          1        1       iitl   maximum number of inner iterations per group until fission source is near
*                                  convergence, i.e. lambda is near convergence.(default=1)
*          0       30       iitm   maximum number of inner iterations per group when close to fission source convergence
*                                  (default calculated)
*        100      100       oitm   maximum number of outer iterations (default=20)
*          0        0       itlim  iteration time limit (seconds)
*
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************


*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                    ...block v -- solver input (continued)...
*
*****************************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*
*                                        ...miscellaneous parameters(array name = misc)...
*
*      0.000E+00 0.000E+00  bhgt   buckling height
*      1.000E+00 1.000E+00  nnrm   normalization factor
*
*          0        0       influx 0/1 no/yes - read input flux from file rtflux (atflux for adjoint)
*          0        0       insorc 0/1 no/yes - read input source from file fixsrc
*          0        1       iquad  -3/-2/1/2/3 - source of quadrature constants (default=1)
*                                    -3 snoon file
*                                    -2 hybrid product set (triangular arrangement)
*                                     1 old twotran built-in set
*                                     2 product set (rectangular arrangement)
*                                     3 card input
*
*
*                                           ...output controls(array name = solout)...
*
*                          0  fluxp  0/1/2   none/isotropic/all moments - flux print
*                          0  xsectp 0/1/2   none/principal/all - macroscopic cross section print
*                          0  fissrp 0/1     no/yes - print final fission source rate
*                          0  sourcp 0/1/2/3 no/as read/normalized/both - print inhomogeneous source
*                          0  angp   0/1     no/yes - print angular fluxes
*                          0  raflux 0/1     no/yes - write angular fluxes to file raflxm or aaflxm(if ith=1)
*                          0  rmflux 0/1     no/yes - write flux moments to file rmflux
*                          0  balp   0/1     no/yes - print coarse mesh balances
*
*
*                                           ...parameters inferred from input arrays...
*
*                                    1  inchi  0/1/2 none/one chi/zonewise chi
*                                    0  isdnrx 0/1/n - none/x density vector/full matrix
*                                    0  isdnry 0/1 no/yes - use y density vector
*                                    0  isan   source anisotropy
*                                    0  isonse number of source moments input
*                                    0  isonsx number of sourcx moments input
*                                    0  isonsy number of sourcy moments input
*                                    0  isonsf number of sourcf moments input
*                                    0  iql    -1/0/1/2 isotropic/none/all angles/vectors -left boundary source
*                                    0  iqr    -1/0/1/2 isotropic/none/all angles/vectors -right boundary source
*                                    0  iqt    -1/0/1/2 isotropic/none/all angles/vectors -top boundary source
*                                    0  iqb    -1/0/1/2 isotropic/none/all angles/vectors -bottom boundary source
*
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
```

```
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*
*                                        ...parameters from block i...
*
*                        106  igeom   106/107  x-y/r-z
*                          1  ngroup  number of energy groups
*                          8  isn     angular quadrature order
*                          5  mt      number of permanent materials
*                          6  nzone   number of zones
*                         24  it      number of fine mesh x intervals
*                         24  jt      number of fine mesh y intervals
*
*
*
*********************************************************************************************************************************
*********************************************************************************************************************************


*********************************************************************************************************************************
*********************************************************************************************************************************
*
*
*                                        ...material assignments to zones...
*
*
*******************************
*key start matls to zones     *
*******************************
*
*                    zone cross section = sum over matls in the zone of (matl cross section)*( c0 + c1*cmod )
*                                              with
*                                          cmod =  0.000000E+00
*
*                           entry    zone        material        c0              c1
*                                   no. name     no. name
*
*                             1    1 coolant    1 cool          1.000000E+00    0.000000E+00
*                             2    2 homofuel   2 fuel          9.500000E-01    0.000000E+00
*                             3    2 homofuel   1 cool          3.000000E-02    0.000000E+00
*                             4    2 homofuel   3 clad          2.000000E-02    0.000000E+00
*                             5    3 control    3 clad          5.000000E-02    0.000000E+00
*                             6    3 control    5 poison        9.500000E-01    0.000000E+00
*                             7    4 ctrlgap    4 fluid         1.000000E+00    0.000000E+00
*                             8    5 ctrltube   3 clad          1.000000E+00    0.000000E+00
*                             9    6 ctrlcool   1 cool          1.000000E+00    0.000000E+00
*
*
*
************************
*key start storage map *
************************
*
*  scm storage summary...
*
*     total scm required for this problem  74926
*     maximum scm available (maxscm= )     89000
*     scm required for transport           51858
*     scm required for diffusion           46554
*
*
*  lcm storage summary...
*
*     lcm selected for this problem              11037
*     maximum lcm specified (maxlcm= )           30000
*          storage required for all quantities  in core is   11037
*
*          diffusion params are in core.  If O.K. to put in core, set MAXLXM .ge.    11037
*          flux moments     are in core.  If O.K. to put in core, set MAXLXM .ge.    11037
*          diffusion fluxes are in core.  If O.K. to put in core, set MAXLXM .ge.    11037
*          scalar tr fluxes are in core.  If O.K. to put in core, set MAXLXM .ge.    11037
*          storage required for all quantities  on disk is   11037
*


      11037 words lcm required lcmadd
*
* zero flux
*
************************
*key start sn constants*
************************
*
* s 8 constants
*                  mu             eta            weight
*
*     1      0.19232747E+00  0.96229948E+00  0.29197117E-01
*     2      0.57735027E+00  0.79352178E+00  0.23313808E-01
*     3      0.19232747E+00  0.79352178E+00  0.23313808E-01
*     4      0.79352178E+00  0.57735027E+00  0.23313808E-01
*     5      0.57735027E+00  0.57735027E+00  0.22525800E-01
*     6      0.19232747E+00  0.57735027E+00  0.23313808E-01
*     7      0.96229948E+00  0.19232747E+00  0.29197117E-01
*     8      0.79352178E+00  0.19232747E+00  0.23313808E-01
*     9      0.57735027E+00  0.19232747E+00  0.23313808E-01
*    10      0.19232747E+00  0.19232747E+00  0.29197117E-01
```

```
**************************************************material map*************************************************
************************
*key start material map*
************************
material map for j mesh intervals  24, to   1 and i mesh intervals   1, to  24.

                5 5 5 5 5 5 5 5
                5 3 3 3 3 3 3 5
                5 3 3 3 3 3 3 5
                5 3 3 1 1 3 3 5
1 1 1 1 1 1 1 1 5 3 3 1 1 3 3 5 1 1 1 1 1 1 1 1
1 1 1 2 2 1 1 1 5 3 3 3 3 3 3 5 1 1 1 2 2 1 1 1
1 2 2 2 2 2 2 1 5 3 3 3 3 3 3 5 1 2 2 2 2 2 2 1
1 2 2 2 2 2 2 1 5 5 5 5 5 5 5 5 1 2 2 2 2 2 2 1
1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1
1 2 2 2 2 2 2 1 1 1 1 1 2 2 1 1 1 1 1 2 2 2 2 2 2 1
1 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 2 1 1 1
1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1
5 5 5 5 5 5 5 5 5 1 2 2 2 2 2 2 1 5 5 5 5 5 5 5 5
5 3 3 3 3 3 3 3 5 1 2 2 2 2 2 2 1 5 3 3 3 3 3 3 5
5 3 3 3 3 3 3 3 5 1 1 1 2 2 1 1 1 5 3 3 3 3 3 3 5
5 3 3 1 1 3 3 5 1 1 1 1 1 1 1 1 5 3 3 1 1 3 3 5
5 3 3 1 1 3 3 5 1 1 1 1 1 1 1 1 5 3 3 1 1 3 3 5
5 3 3 3 3 3 3 5 1 1 1 2 2 1 1 1 5 3 3 3 3 3 3 5
5 3 3 3 3 3 3 5 1 2 2 2 2 2 2 1 5 3 3 3 3 3 3 5
5 5 5 5 5 5 5 5 1 2 2 2 2 2 2 1 5 5 5 5 5 5 5 5
                1 2 2 2 2 2 2 1
                1 2 2 2 2 2 2 1
                1 1 1 2 2 1 1 1
                1 1 1 1 1 1 1 1

*****************************************************************************************************************
*****************************************************************************************************************
*
*                    ...cross section related data from file macros  03/17/11:05: version  1 ...
*
*****************************************************************************************************************
*
*
*  1 cool      2 fuel      3 clad      4 fluid      5 poison
*
*
*****************************************************************************************************************

*****************************************************************************************************************
*****************************************************************************************************************
*
*                                  ...iteration controls and criteria...
*
*****************************************************************************************************************
*
*
*                                        ***iteration criteria***
*
*                                             transport inners
*
*         criterion    quantity to test                                  value     action taken if value exceeded
*
*            iitl  - inner iteration count until near lambda              1        terminates inners
*                      (i.e. fission source) convergence
*            iitm  - inner iteration count when near lambda              30        terminates inners
*                      (i.e. fission source) convergence
*            epsi  - fractional ptwise flux change                    1.00E-04    does another inner
*                      per inner
*
*                                          diffusion sub-outers
*
*         criterion    quantity to test                                  value     action taken if value exceeded
*
*            oitmd - sub-outer iteration count                           100        terminates sub-outers
*            eps   - diffusion lambda-1.0 (see note below)            1.00E-04    does another sub-outer
*            eps   - fractional ptwise fission change                 1.00E-04    does another sub-outer
*                      per sub-outer (see note below)
*
*            note: eps, when the problem is finally converged, will equal epsi, the value shown above. however,
*                  early in the iteration process, a larger value may be used to avoid unnecessary iterations.
*
*                                        final convergence criteria
*
*         criterion    quantity to test                                  value     action taken if value exceeded
*
*            oitm  - outer iteration count                               100        quits with error message
*            epsi  - transport lambda-1.0                             1.00E-04    does another outer
*
*****************************************************************************************************************
*****************************************************************************************************************
```

```
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                    ...flux and eigenvalue convergence as monitored by twodant...
*
*********************************************************************************************************************************
*****************************
*key start iteration monitor *
*****************************
*
*
* cpu time outer       diffusion     k-eff                  max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  eigenvalue   lambda-1     flux change  fiss change converged
*  3.45     0
*  4.13     1      0       2      0.99418965  4.74215E-03  0.00000E+00  1.70363E-01  **no**
*  4.30     2      2       3      1.00981730  1.48191E-02  2.03574E-01  1.42439E-02  **no**
*  4.49     3      2       5      1.01026595  1.01339E-04  5.43373E-02  4.36630E-03  **no**
*  4.76     4      2      12      1.01081947  4.50688E-04  1.08803E-02  1.51584E-03  **no**
*  4.93     5      2       3      1.01072683 -1.17141E-04  5.84303E-03  6.13913E-04  **no**
*
*    ------------------------------------------------------------
*
*             -- inner iteration summary for outer iteration no.   6 --
*
*                                iter per max flux   at
*                           group  group   change   mesh
*                             1      7   0.74E-04  22,  7
*
*
* cpu time outer       diffusion     k-eff                  max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  eigenvalue   lambda-1     flux change  fiss change converged
*  5.51     6      7       10     1.01074723  9.76594E-06  7.44830E-05  9.40299E-05   yes
*
*               $$$$$$ all convergence criteria satisfied $$$$$$
*
* particle balance = -1.92403E-06      total inners all outers =   15
*
*********************************************************************************************************************************

*
*
*********************************************************************************************************************************
*********************************************************************************************************************************
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                        ...group edit and balances upon convergence...
*
*****************************************************************************************************************************************
*
*
*                       ...title---        twodant/gq test case                                              ...
*
*
*                                        ...system balance tables...(neutrons only)
*
*
*******************************
*key start balance table     *
*******************************
*
*
*        gp          source      fission source          in scatter        self scatter         out scatter          net leakage
*
*         1   0.0000000E+00        1.0000000E+00       0.0000000E+00        3.5451763E+00       1.4210855E-14        2.0156399E-06
*
*       tot   0.0000000E+00        1.0000000E+00       0.0000000E+00        3.5451763E+00       1.4210855E-14        2.0156399E-06
*
*
*
*        gp       absorption    particle balance        right leakage  horizontal leakage         top leakage     vertical leakage
*
*         1   9.9999991E-01       -1.9240310E-06       -7.6133049E-05        1.0366833E-06        6.8428274E-03        9.7895660E-07
*
*       tot   9.9999991E-01       -1.9240310E-06       -7.6133049E-05        1.0366833E-06        6.8428274E-03        9.7895660E-07
*
*
*
*        gp     left leakage      bottom leakage      nprod spectrum
*
*         1   7.7169732E-05       -6.8418484E-03        1.0000000E+00
*
*       tot   7.7169732E-05       -6.8418484E-03        1.0000000E+00
*
*****************************************************************************************************************************************
  Timing data...tord, tswep, tdsa =    0.6    0.8    0.6 CPU seconds.


integral summary information
summary integral-k-eff     1.0107472E+00
integral-source-i             neutron  0.0000000E+00
integral-fission-i            neutron  1.0000000E+00
integral-in-scak-i            neutron  0.0000000E+00
integral-self-scak-i          neutron  3.5451763E+00
integral-out-scak-i           neutron  1.4210855E-14
integral-net lkage-i          neutron  2.0156399E-06
integral-absorption-i         neutron  9.9999991E-01
integral-right lkage-i        neutron -7.6133049E-05
integral-horizontal lkage-i   neutron  1.0366833E-06
integral-top lkage-i          neutron  6.8428274E-03
integral-vertical lkage-i     neutron  9.7895660E-07
*
*
*                                        ...interface file rtflux written..

*
*
*                                        ...interface file sncons written..


twodant iteration time, mins  9.2918E-02
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
```

```
                                    run highlights

*key start run highlights    *

                          ****************************************
                          *                                      *
                          *    all modules are tentatively go.   *
                          *     interface file geodst written.   *
                          *       cross sections from cards.     *
                          *    interface mixing files written.   *
                          *     interface file asgmat written    *
                          *    xs files macxs,smxedt written.    *
                          *     interface file solinp written.   *
                          *   edit module execution suppressed.  *
                          *  neither editit nor edit cards exist.*
                          *        start solver execution.       *
                          *      all convergence criteria met.   *
                          *     interface file rtflux written.   *
                          *     interface file smcons written.   *
                          *                                      *
                          ****************************************

                                storage and timing history

                          ************************************************************
                          *        scm     scm     lcm     lcm     cpu     sysio     *
                          * module words   limit   words   limit   seconds seconds   *
                          *  ------ ------  ------  ------  ------  ------  ------    *
                          *      0      0       0       0       0     5.6     0.6    *
                          *    100  88949   89000       0       0     3.3     0.3    *
                          *    101      0       0       0       0     0.7     0.0    *
                          *    102  88949   89000     477   30000     0.2     0.0    *
                          *    103     36   89000       0       0     0.0     0.0    *
                          *    104   2084   89000       0       0     0.1     0.0    *
                          *    105      0       0       0       0     0.0     0.0    *
                          *    106      0       0       0       0     0.0     0.0    *
                          *    107    184   89000      35   30000     0.1     0.0    *
                          *    108    606   89000       3   30000     0.0     0.0    *
                          *    109      0       0       0       0     0.0     0.0    *
                          *    112      0       0       0       0     0.0     0.0    *
                          *    200  88947   89000   11037   30000     2.3     0.3    *
                          *    201  88947   89000       0       0     0.1     0.1    *
                          *    202      0       0       0       0     0.0     0.0    *
                          *    203   4539   89000       0       0     0.0     0.0    *
                          *    204      0       0       0       0     0.0     0.0    *
                          *    205      0       0       0       0     0.0     0.0    *
                          *    206      0       0       0       0     1.7     0.1    *
                          *    207      0       0       0       0     0.4     0.0    *
                          *    208      0       0       0       0     0.0     0.0    *
                          *    210      0       0       0       0     0.0     0.1    *
                          *    211      0       0       0       0     0.0     0.1    *
                          *    400      0       0       0       0     0.0     0.0    *
                          ************************************************************

                                   ...execution terminated...
```

APPENDIX B: OPERATING SYSTEM SPECIFICS

UNIX/UNICOS Execution

On UNIX or UNICOS systems, the input is on STDIN and the printed output is on STDOUT. Thus, the user will normally cause execution of the program with the command:

dant.x < *odninp* > *odnout*

where dant.x is the name of the executable file, *odninp* is the user's choice for a name for the input file, and *odnout* is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is dant.x.

STDERR contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDERR are any error messages.

Library Search Path

Most files read or written by TWODANT/GQ are in the current UNIX working directory. Some forms of cross-section files may be kept in other directories. By setting the environment variable SNXSPATH, the user may specify an ordered set of alternate directories in which the program should look for the named files. As an example, if an ISOTXS file is in the directory, **/usr/tmp/xs**, then the following command can be used

```
setenv SNXSPATH /usr/tmp/xs
```

and TWODANT/GQ will then look in that named directory for the library. The search path for each of the possible libraries is given in Table 4.2.

Table 4.2 UNIX Search Path

LIB	SEARCH PATH
MACRXS	Current Working Directory (CWD).
GRUPXS	**SNXSPATH**, then CWD.
ISOTXS	**SNXSPATH**, then CWD.
BXSLIB	**SNXSPATH**, then CWD, but see text below.
ODNINP	None, the library is contained in the input file.
MACBCD	CWD
XSLIBB	CWD
MENDF[a]	Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.
MENDFG[b]	
XSLIB	**SNXSPATH**, then CWD
other	For any name other than those above, the program will assume the form is XSLIB and search for it in **SNXSPATH**, then CWD.

a. Available only at Los Alamos.

b. Available only at Los Alamos.

SNXSPATH can be used to protect an input BXSLIB file from being overwritten. See the discussion on page 4-49.

TWOHEX USER'S GUIDE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

5

XTM — Transport Methods Group

Los Alamos
National Laboratory

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

An Affirmative Action/Equal Opportunity Employer

This work was supported by the US Department of Energy.

USER'S GUIDE FOR TWOHEX: A CODE PACKAGE FOR TWO-DIMENSIONAL, NEUTRAL-PARTICLE TRANSPORT IN EQUILATERAL TRIANGULAR MESHES

by
Wallace F. Walters, Forrest W. Brinkley,
and Duane R. Marr

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The TWOHEX code is a modular computer program designed to solve the two-dimensional, time-independent, multigroup discrete-ordinates form of the Boltzmann transport equation.

TWOHEX™ is based on the modular construction of the DANTSYS™ code system package. This modular construction separates the input processing, the transport equation solving, and the postprocessing, 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 used by all the codes in the package. With these modules, production codes with different solution techniques are invoked simply by executing different SOLVER modules in the package. This SOLVER choice is automatically made by the package through an analysis of the input stream.

The TWOHEX code is then simply the DANTSYS package with a two-dimensional SOLVER module based on an equilateral triangular mesh.

Some of the major features included in the TWOHEX package are:

1. a free-field format 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 standard Chebychev acceleration[2,3] of both the inner and outer iterations,
4. direct (forward) or adjoint calculational capability,
5. geometry domains for sixth core, third core, and whole core including appropriate boundary conditions,
6. arbitrary anisotropic scattering order,
7. inhomogeneous (fixed) source or k_{eff} calculation options,
8. nodal/characteristic transport method for solution[4] of the transport equation,

TWOHEX and DANTSYS are trademarks of the Regents of the University of California, Los Alamos National Laboratory.

9. user flexibility in using either ASCII text or sequential file input.

10. user flexibility in controlling the execution of both modules and submodules, and

11. extensive, user-oriented error diagnostics.

DOCUMENTATION FOR TWOHEX USAGE

The documentation described here constitutes a complete manual for the use of the TWOHEX code. It is intended to fully replace the former TWOHEX manual.[5]

Included are two general categories of information. The first category is in this User's Guide and is oriented towards preparing input to the code. The second category is of a background, reference, conceptual, or theoretical nature and is intended primarily for the novice or first time user; an experienced user generally needs only this User's Guide.

What Is In This User's Guide

This User's Guide is a chapter from the much larger DANTSYS document. This Guide provides the ASCII text input specifications for TWOHEX.

The guide is intended to serve as a complete input manual for two classes of user. Special, succinct sections containing summaries and compact tables are intended for the advanced user in order to make his input preparation more efficient. The main body of the guide concerns itself with descriptions of the input and should be sufficient for the user familiar with discrete ordinates concepts. Novice users may find other chapters of the document necessary.

This Guide first gives an overview of the input block order required by the code.

Next is a "mini-manual" in which are listed all the names of available input arrays arranged by input block. Definitions of input arrays are not given, as the names are suggestive, but expected types and sizes are provided. This mini-manual is very useful to the user as a quick check for completeness, a quick reference to type and size, and as an index into the more detailed array descriptions that follow. For the experienced user, the mini-manual is frequently all that is needed to prepare a complete input deck.

Following the mini-manual are reference sections describing in detail all the input parameters and arrays.

Appendix A provides a sample TWOHEX case with model, input, and output descriptions.

Lastly, Appendix B details operating system specifics, including how to effect an execution of the code.

Information of a reference, background, or theoretical nature that the first time user may need may not be found in this User's Guide, but the user will encounter liberal references to other chapters of this document for that sort of information.

What Is Available Elsewhere

In addition to this User's Guide, the user, especially the first time user, may find the information below described in other chapters of this document pertinent. For even greater detail on some of the general items, particularly the methods items, the user should look at Ref. 6.

The chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1 discusses in more detail the geometry and solver concepts and their related input. If the User's Guide proves insufficient for your needs, look in this chapter. As TWOHEX is not as fully featured as the other codes in the package and does not use diffusion acceleration, some of the information there is inappropriate for TWOHEX. A useful section there is the one on the input of inhomogeneous sources. There is also more detail on the Block-I input.

A discussion of how the EDIT module works and more detail on preparing the input is given in the chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 serves as the reference manual for the free-field input (rules, format, and operators) used in this code. That chapter is summarized in this guide, but should the summary prove inadequate, the user is referred there for full details.

The chapter "CROSS-SECTION LIBRARIES" starting on page 10-1 gives details of the many library formats available to TWOHEX, including sections on how to prepare your own card-image (or text) libraries.

The chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1 describes the mixing concepts in detail and shows some examples.

Next is the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. That chapter describes the theoretical basis for the TWOHEX code as well as the other codes in the DANTSYS package.

In the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure" starting on page 13-1 is shown a brief 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. In particular, the code package structure must be understood in order to make up input for piecewise executions of the code that are possible with controls that are part of the input in Block-I.

Error diagnostics that the user might encounter are found in the chapter "ERROR MESSAGES" starting on page 14-1. Several examples of input errors and the resulting error messages are provided for the user.

The chapter "FILE DESCRIPTIONS" starting on page 15-1 is a reference that describes all the files used by the package. Included is a detailed description of the file structure of

the code dependent, binary, sequential interface files generated by and used in the DANTSYS package. Also included are descriptions of any other files produced or used by the package, both binary and text. In some cases, this may simply be a reference to a more comprehensive document, such as the file descriptions for the CCCC standard interface files.

TWOHEX INPUT OVERVIEW

Input Block Order

The full TWOHEX input consists of a title section, followed by six blocks of free field input. The title section is not free field. Any input referred to as a block uses the free field input form.

Block-I consists of basic control and dimensional information that allows efficient packing of the array data. This information also allows checking of the lengths of arrays supplied by interface files.

Block-II contains the geometric information

Block-III consists of the nuclear data specifications.

Block-IV contains mixing information.

Block-V contains the rest of the input needed for specifying the flux calculation.

And lastly, Block-VI contains the edit (i.e., report writing) specifications.

If a text cross-section library is to be included in the input deck, it should be placed between Blocks III and IV. TWOHEX supports many library formats and so the library may or may not be in free field format depending upon the option chosen.

A full input would then look like that diagrammed in Figure 5.1 on the following page.

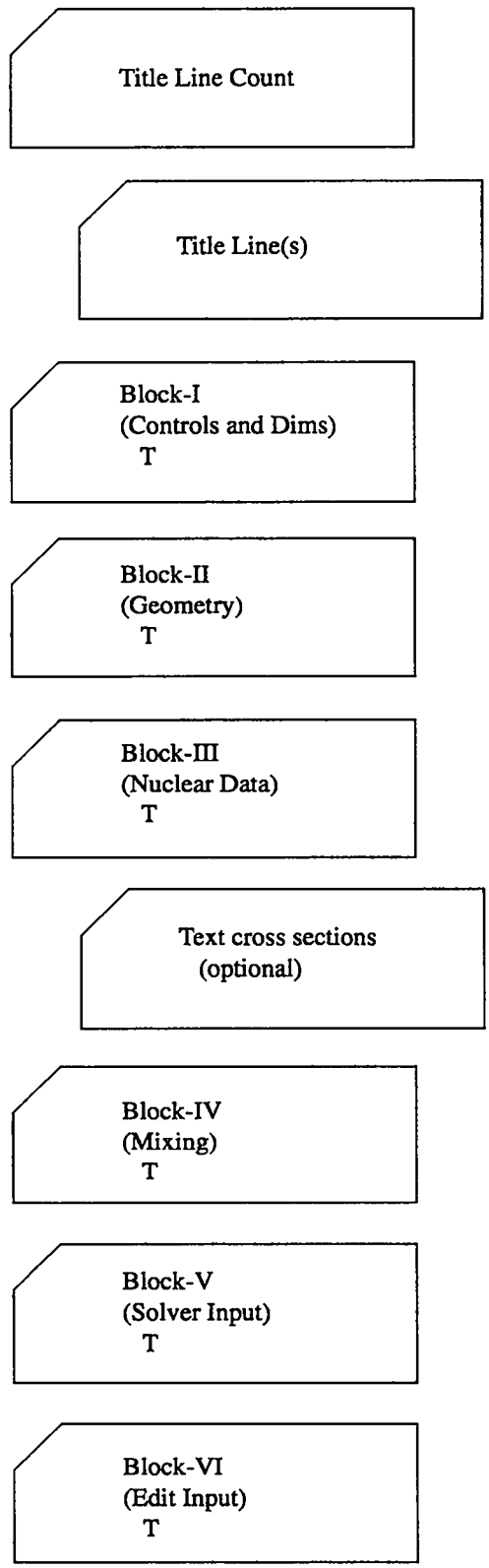


Figure 5.1 TWOHEX Input Order

Free Field Input Summary

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 is summarized here for quick reference.

There are four basic input quantities in the free field input used in TWOHEX; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is briefly described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array." To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, might look like:

SOURCX= 0 0 0 1.1 1.1 0 0 0 0 0

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Single value input variables are treated as arrays of unit length.

Numeric Data Items

Numeric data items follow a Fortran input convention. For example, all of the following are valid entries for the number ten:

10, 1.0+1, 1E1, 10.0

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a Fortran variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters and blanks may be included if the data item is surrounded by double quotes. Operators may NOT be used with character data items.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this, the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described later, in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read, no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input.

The data operators are specified in the general form

n *O* d

where:

n is the "data numerator", either an integer or a blank;
O is any one of the "data operator" characters shown below; and
d is a "data entry" (may be blank for some operators).

Note: The "data operator" character must be appended to the "data numerator".

Using operators, the SOURCX input described above could more succinctly be given as:

SOURCX= 0 0 0 2R 1.1 F0

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO[7] operators may be used in numeric entry.

A table of the most used operators is given next including brief descriptions. For full descriptions of these and a complete list of all the available operators, including the more esoteric ones, the user is referred to "FREE FIELD INPUT DETAILS" on page 9-13.

Frequently Used Operators

Operator[a]	Functionality
nR d	REPEAT the data item d, n times.
nI d	INTERPOLATE (linear) n data items between data item d and the next data item.
nC d	SCALE (multiply) the n previous entries by d.
F d	FILL the rest of the data string with the data item d.
nY m	STRING REPEAT. Repeat the previous m strings, n times.
nL d	INTERPOLATE LOGARITHMICALLY n data items between d and the next d.
nZ	ZERO. Enter the value zero n successive times.
nS	SKIP. Skip the next n data items.
nQ m	SEQUENCE REPEAT. Enter the last m entries, n more times.
nG m	SEQUENCE REPEAT WITH SIGN CHANGE. Same as the Q option but the sign of the m entries is changed every repeat.
nN m	SEQUENCE REPEAT INVERT. Same as the Q option but the order of the m entries is inverted each repeat.
nM m	SEQUENCE REPEAT INVERT WITH SIGN CHANGE. Same as N option but the sign is also changed every repeat.
nX	COUNT CHECK. Causes code to check the number of entries in the current string so far, against the number n.

a. The operator character must always be appended directly to n. d or m need not be immediately adjacent to the operator character.

MINI-MANUAL Introduction

On the following few pages is given a complete list of the input names, expected array sizes, and order within the array. No description of the array contents is given in this MINI-MANUAL as full details are given in later sections. The MINI-MANUAL is intended to serve as a quick reference for the knowledgeable user.

In both the MINI-MANUAL and in the detailed sections which follow, a shorthand form is used to indicate the size and order of the array that the code expects. This information is enclosed in square brackets immediately after the array name. Essential features are:

1. A single entry in the brackets is the array length.

2. No brackets at all indicates a simple variable (i.e., an array of unit length).

3. A dash (-) in the brackets indicates an arbitrary length.

4. A semicolon (;) indicates that the input for that array is expected in strings. To the left of the semicolon is the string length. To the right of the semicolon is the number of strings in the array.

5. If the number of strings is shown as a product, the order is important. The left-most quantity must be exhausted first, then, the next one to the right is varied. For example, the array name for the full spatial source distribution is shown as:

   SOURCF [IT;JT*NMQ]

   where - IT is the number of meshes in the X-direction, JT is the number of meshes in the Y-direction, and NMQ is the number of input source moments. For this array, the first string is composed of the P_0 source values for each x mesh point in the first y mesh. The next string is the P_0 source values in the second y mesh. This process is repeated for all JT y meshes. Then starting again with the first y mesh, the P_1 source values for each x mesh are given. After all P_1 values are given, the P_2 values follow. Continue until all NMQ moments are specified.

Note: Usually, values for the quantities within brackets will have already been specified in the input. Sometimes, however, a quantity is derived from the array input itself. For instance, in this particular case, NMQ is not an input quantity; rather, the code counts the number of strings and then, knowing JT, deduces what NMQ must have been.

MINI MANUAL

Title Line Control
(3I6 Format)
NHEAD,NOTTY,NOLIST

Title Line(s)

(IF NHEAD>0)

Block-I:Controls & Dimensions

IGEOM
NGROUP
ISN
NISO
MT
NZONE
IT
JT

MAXLCM
MAXSCM

-- -- -- -- --

NOSOLV
NOEDIT

NOGEOD
NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM

T

Block-II:Geometry

DOMAIN valid: THIRD
SIXTH
WHOLE
HEIGHT
ZONES [IM;JM]

BSQ [1]
-or-
BSQ [NZONE;NGROUP]

T

Block-III: Cross Sections

LIB
valid: *ODNINP*
XSLIB
ISOTXS
GRUPXS
BXSLIB
MACRXS
MACBCD
XSLIBB
(local)*MENDF*
(local)*MENDFG*
alternate XSLIB name

WRITMXS
valid: *MACBCD*
XSLIBB
XSLIBF
XSLIBE

LNG
BALXS
NTICHI
CHIVEC [NGROUP]
LIBNAME
-- -- -- -- -- -- --
Rest of this block is needed only for text libraries.

MAXORD
IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD (default:1)
NAMES [NISO]
EDNAME [IHT-3]
NTPI [NISO]
VEL [NGROUP]
EBOUND [NGROUP+1]

T

iff LIB= ODNINP, insert
ASCII text cross sections here

Block-IV: Mixing

MATLS [-;MT]
ASSIGN [-;NZONE]

PREMIX [-;-]

MATSPEC [-]
valid: ATFRAC
WTFRAC
ATDEN

ATWT [-]
MATNAM [MT]
ZONNAM [NZONE]
T

Block-V: SOLVER

IEVT
ISCT
ITH

EPSI
OITM
ITLIM

FLUXP
XSECTP
FISSRP
SOURCP

--- Flux Guess -------

INFLUX

--- Quadrature ---

IQUAD

--- Miscellaneous ---
NORM
BHGT

CHI [NGROUP;M]

DEN [IT;JT]
-or-
DENX [IT], DENY[JT]

----Volumetric Source----

INSORS
SOURCE [NGROUP;NMQ]
-or-
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ]
-or-
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ] and
SOURCE [NGROUP;NMQ]
-or-
SOURCF [IT;JT*NGROUP*NMQ]
-or-
SOURCF [IT;JT*NMQ]
and SOURCE [NGROUP;NMQ]
T

Block-VI: EDIT

PTED
ZNED

POINTS [K], K≤IT*JT
EDZONE [IT;JT]

EDXS [K], K≤NEDT
RESDNT
EDISOS [K], K≤NISO
EDCONS [K], K≤NISO
EDMATS [K], K≤MT
XDF [IT]
YDF [JT]

RSFE [NGROUP;-]
RSFX [IT;-]
RSFY [JT;-]
RSFNAM [-]

ICOLL [K], K≤NGROUP
IGRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

T

TWOHEX INPUT DETAILS

Introduction

The following pages of this section give details for each of the input arrays. All valid TWOHEX arrays are discussed in this section in detail complete enough to form the input.

However, the beginning user, particularly one unfamiliar with discrete-ordinates codes, may find that he is missing some information of a background nature. See "What Is Available Elsewhere" on page 5-14 for that.

First, here are a few general instructions:

1. All six of the input blocks are normally included. Block-I is always required but any of the other five blocks may be omitted under the proper conditions. The input module reads each block in turn and from it generates one or more binary interface files. The interface files drive the SOLVER and EDIT modules. Thus, if the user wants no edits, the Block-VI input may be omitted. Then with no interface file, the EDIT module will not be executed. Alternatively, if the interface file is available from another source, the corresponding block of input may be omitted. For instance, Block-II describes the geometry. The input module normally writes this information to the GEODST interface file. If the GEODST file is available from another source or a previous run, the Block-II input may be omitted.

2. A general theme of the TWOHEX input is that arrays that are not needed are not entered. Presence of an array indicates that it should be used. Thus, for example, if the density array is entered (DEN array), the cross section at each mesh interval will be modified accordingly. No separate switch need be set to say that the calculation should be done. To eliminate the density modification, simply remove the DEN array from the input or comment it out.

3. The arrays, in general, are grouped in the input instructions according to function. Thus, for example, the input arrays for the volumetric source are found in a single table, or grouping, of input.

4. Groupings of input data may be marked as "Required" or "Optional" in order to guide the user and speed navigation through the input instructions.

   "Required" means that at least one of the arrays in the grouping must be entered. Thus, you must read through the grouping and enter at least one of the arrays found there.

Groupings marked "Optional" may be skipped if the subject is inappropriate. Thus, using the previous example, if one has no volumetric source, one simply skips to the next grouping of input; there is no need to read about any of the arrays within the volumetric source grouping.

Arrays in groupings not marked as "Required" or "Optional" should be reviewed. These groupings contain arrays of vital data that are used in every calculation, but have default values. Thus, although you may not make any input to these arrays and they are in that sense optional, you must concern yourself with them to ensure that the default values are what is intended.

5. Input arrays may also be marked individually. If not marked, they inherit the marking of the grouping in which they are contained. Thus, an unmarked array in a "Required" grouping is required input and you must enter that array. An unmarked array in an "Optional" grouping is optional.

   You may encounter a "Required" array within an "Optional" grouping. That means that if you decide to invoke the option represented by that grouping, you must input that particular array. For example, if you want user defined response function reaction rates calculated, you must input the RSFE array.

   All arrays within unmarked groupings are optional. However, values in these arrays may be used by the code, so you should concern yourself with the default values if you choose not to enter a value.

6. Unless specifically noted otherwise, the default on all numeric inputs is zero.

7. In an adjoint run, none of the groupwise input arrays should be inverted. The code will externally identify all groups by the physical group number, not by the calculational group number (the calculational group number is in inverse order). Thus, the user interface should be consistently in the physical group order.

8. The use of information within square brackets to indicate the size of arrays and strings and the order within those arrays is the same as described in "MINI-MANUAL Introduction" on page 5-22.

9. Except where noted, arrays and strings must contain the exact number expected by the code (as indicated in the array or string description). If not, the code will eventually abort with a (hopefully) descriptive error message or messages.

10. New users reading these instructions for the first time and unfamiliar with the TWOHEX input may find it helpful to follow the sample input in Appendix A while reading this section.

11. Array names are shown here in upper case. What you should actually input for them will depend upon the code's implementation on your platform. At the present time, on most platforms, you should use lower case input.

12. Items in italics in the input instructions indicate actual values that may be entered for an array. You will frequently find switches where the input is the digit 0 or the digit 1. This will be represented by *0/1* in the input description. In other arrays where an exact character string is required such as "ISOTXS" in the LIB array, you will find the notation *ISOTXS*. Note that in this notation the word is both upper case and italicized. This combination means you must enter exactly those characters. Again, although the characters will be shown here in upper case, what you should actually input for them will depend upon the code's implementation on your platform.

13. When a template for the input form is given, as for the MATLS array, the style in the template tells the user what is expected. If an input word or value is lower case and italicized, the user is to replace that position with the entry of his choice. If the input word is in italicized style and in upper case, the user is to input exactly those characters to achieve the desired result. Depending on the implementation on your platform, the input word, itself, is usually in lower case.

14. Units to be used for the input quantities are not spelled out as they only need to be self consistent. However, the following are commonly used: Dimensions in centimeters, isotopic cross sections in barns per atom; then it follows that atom densities are in atoms per barn-centimeter. Sources are particles per cm^3 per second for volumetric sources and particles per cm^2 per second for boundary sources; fluxes will then be in particles per cm^2 per second.

Title Line Details

Title Line Control (format 3I6)[a] {Required}

Word	Name	Comments
1	NHEAD	Number of title lines that follow.[b]
2	NOTTY	Suppress output to on-line user terminal? *0/1* = no/yes.
3	NOLIST	Suppress listing of all ASCII text input? *0/1* = no/yes. (default=no)

a. WARNING! Note that this first line is in fixed format.

b. Follow this control line with NHEAD title lines containing descriptive comments. Each title line may contain up to 72 characters.

Block-I Details: Dimensions and Controls

Dimensions {Required}

Name	Comments
IGEOM	Geometry. Enter 9 for equilateral triangles or the following character string: IGEOM= HEX
NGROUP	Number of energy groups.
ISN	S_n order to be used. If ISN is negative, code will use the Chebychev-Legendre (IQUAD=2) quadrature set. [See IQUAD input on page 5-47]
NISO	Number of physical isotopes on the basic input cross-section library.
MT	Number of physical materials[a] to be created.
NZONE	Number of geometric zones[b] in problem.
IT	Total number of triangles in each horizontal band.
JT	Total number of bands (must be IT/2 for THIRD or SIXTH core domains).

a. Material is defined on page 5-41.

b. Zone is defined on page 7-13.

Storage Requirements {Optional}

Name	Comments
MAXSCM	Length of SCM desired (default=40000_{10})
MAXLCM	Length of LCM desired (default=140000_{10})

The above input (Dimensions plus Storage Requirements) for Block-I will cause the code to attempt to produce a full run, subject to availability of the input normally found in the other Blocks. The controls below allow shortened print files, partial runs (say, of only the input module), or cause the code to ignore any of the other input Blocks present. For full details on their use, see "PIECEWISE EXECUTION" on page 13-19.

Run Configuration Controls {Optional}

Name	Comments
NOSOLV	Suppress solver module execution. *0/1* = no/yes.
NOEDIT	Suppress edit module execution. *0/1* = no/yes.
NOGEOD	Suppress writing GEODST file even though the geometry input (Block-II) may be present. *0/1* = no/yes.
NOMIX	Suppress writing mixing files even though the mixing input in Block-IV may be present. *0/1* = no/yes.
NOASG	Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. *0/1* = no/yes.
NOMACR	Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. *0/1* = no/yes.
NOSLNP	Suppress writing the SOLINP file even though Block-V may be present. *0/1* = no/yes.
NOEDTT	Suppress writing the EDITIT file even though Block-VI may be present. *0/1* = no/yes.
NOADJM	Suppress writing the ADJMAC file even though an adjoint calculation is called for. *0/1* = no/yes.

Note: Default on all these controls is no.

Block-II Details: Geometry

Geometry Arrays {Required}

Name	Comments[a]
DOMAIN	Specifies the orientation and boundary conditions of the solution domain. Enter as a character data item one of the following five character words. Word — Description *THIRD* — Solution domain is a rhombus with a 120 degree angle at the lower left corner, that is, at mesh point (1,1). There are rotational boundary conditions on the left and bottom, vacuum on the top and right. *SIXTH* — Solution domain is a rhombus with a 60 degree angle at the lower left corner, that is, at mesh point (1,1). There are rotational boundary conditions on the left and bottom, vacuum on the top and right. *WHOLE* — Solution domain is a rhombus with a 60 degree angle at the lower left corner, that is, at mesh point (1,1). There are vacuum boundary conditions all around.
HEIGHT	Specifies the height of a single triangle.
ZONES [IM;JM]	Zone number[b] for each triangle. This array defines the geometric zones to which cross-section materials are assigned. The zone number must not be greater than NZONE.
BSQ [1] -or- BSQ [NZONE;NGROUP] {optional}	Buckling to use at every triangle or the buckling by zone and group. If the number of strings entered is less than NGROUP, then the last string entered will also be used for all omitted groups. CAUTION: Subsequent input of a buckling height via the BHGT array in the SOLVER block input will override the BSQ input.

a. The information entered in this block is written to the CCCC standard interface file GEODST. See note on units on page 5-29.

b. A zone number of zero indicates the mesh contains a void, and no cross section will be associated with that mesh. The zero zone number is not counted in the total zone count NZONE. .

Block-III Details: Nuclear Data

Nuclear Data Type and Options {Required}

Name	Comments
LIB	Name[a] and form of the cross-section data file. Enter as a data item one of the following words:
	Word — Description
	ISOTXS[b] — CCCC standard isotope ordered binary cross-section file.
	XSLIB — ASCII text library supplied in a separate file named XSLIB.
	ODNINP — ASCII text library follows after this block of input (after the T of Block-III).
	GRUPXS[c] — CCCC standard group ordered cross-section file.
	BXSLIB — Binary library supplied as a separate file named BXSLIB. [See "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 10-12.
	MACRXS[d] — Use existing files named MACRXS for SOLVER module, SNXEDT for EDIT module. These files were created in a previous run. 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.
	XSLIBB — See "XSLIBB Card-Image Library File" on page 10-12.
	MACBCD — ASCII form of MACRXS file.
	MENDF — (LANL only) See "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.
	MENDFG — (LANL only) See "The Los Alamos MENDF5G Gamma Cross-Section Library" on page 10-14.
	other — If a word other than those listed above is entered, the code will use the file with that word as its name, provided that file exists in the user's file space. Such a file must be structured as an XSLIB file.
WRITMXS {optional}	Controls the code's writing certain ASCII cross-section files.[e] Enter one of the following words:

Nuclear Data Type and Options (Cont.) {Required}

Name	Comments	
	Word	Description
	MACBCD	Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file.
	XSLIBB	Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file.
	XSLIBE	Creates the cross-section file named XSLIBE, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBE is in Los Alamos 6E12 format (IFIDO=0).
	XSLIBF	Creates the cross-section file named XSLIBF, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBF is in FIDO fixed-field format (IFIDO=1).
LNG {optional}	Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.	
BALXS {optional}	cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using!	
	Value	Description
	-1	balance cross sections by adjusting absorption cross section.
	0	do not balance cross sections. (default)
	1	balance cross sections by adjusting self-scattering cross section.
NTICHI {optional}	MENDF fission fraction to be used for the problem (LANL only). *1/2/3* = Pu239/U235/U238 (default is U235). Will be overridden by any CHIVEC input described below or by any zone-dependent CHI in input Block-V.	
CHIVEC [NGROUP] {optional}	Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.	

a. On UNIX systems, the user may specify a search path for some of these files using the environment variable SNXSPATH. See "Library Search Path" on page 5-76 for details.

b. The CCCC standard for file ISOTXS does not allow the inclusion of the 2L+1 term in the higher order scattering cross section. However, if you have a nonstandard file which contains the 2L+1 term, you may override by setting I2LP1=1. See "Text Cross-Section Library Format" on page 5-38. TWOHEX will then convert the cross sections to the appropriate internal form.

c. The 2L+1 term on GRUPXS is treated the same as for ISOTXS. See footnote b.

d. In the convention used in this user's guide, a MACRXS library contains "material" cross sections; all the other libraries contain "isotope" cross sections.

e. See "COUPLED NEUTRON-GAMMA CROSS SECTIONS" on page 10-15.

Alternate Library Name {Optional}

Name	Comments
LIBNAME	Alternate name of the library file. May be used only with certain types of libraries. See Table 5.1.

The entries in the LIB input variable normally dictate both the form and the name of the cross section library. If the user specified ISOTXS, for example, the code would look for a file named ISOTXS and expect it to be in the CCCC format for an ISOTXS file.

For some libraries, the user may specify the form in the LIB array and specify separately the name in the LIBNAME array. The libraries that can be treated this way are shown in Table 5.1.

Table 5.1 LIBNAME Availability

LIB	LIBNAME AVAILABLE?
MACRXS	No
GRUPXS	Yes
ISOTXS	Yes
BXSLIB	Yes
ODNINP	No
MACBCD	No
XSLIBB	No
MENDF[a]	No
MENDFG[b]	No
XSLIB	Yes
other	Ignored

a. Available only at Los Alamos.

b. Available only at Los Alamos.

The BXSLIB file requires special treatment. It is normally created when the original library is a text library in the ODNINP or XSLIB form. In subsequent runs, this binary BXSLIB file may be used as the source of the cross-section data. The user may wish to save this file under another name. The program, in future runs, may then access the library for reading by using LIBNAME to specify that name.

This procedure is wise because some cases using the BXSLIB form as input also require rewriting it in order to add new information. When this situation arises, the rewritten file is always named BXSLIB. Thus, if the original BXSLIB form library had a different name, it would be protected from being overwritten. For the remainder of the current run, the program will access the file named BXSLIB.

Text Cross-Section Library Format {Required if LIB= XSLIB or LIB=ODNINP}

Name	Comments
MAXORD	Highest Legendre order in the scattering tables.
IHM	Number of positions (entries) in each row of the cross-section table.
IHT	Position number of the total cross section.
IHS {optional}	Position number of the self-scatter cross section. (default = IHT + 1).
IFIDO {optional}	Format of the cross-section library. *-1/0/1/2* = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.
ITITL {optional}	A title line precedes each table. *0/1* = no/yes
I2LP1 {optional}	Higher order scattering cross sections on the library contain the 2L+1 term. *0/1* = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.
SAVBXS {optional}	Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. *0/1* = no/yes.
KWIKRD {optional}	Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? *0/1* = no/yes (default=yes).
NAMES [NISO] {optional}	Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).
EDNAME [IHT-3] {optional}	Character name for each of the EDIT cross-section positions used in the cross-section edits. These are the positions before the absorption cross section in the cross-section table. (default names are: EDIT1, EDIT2, . . .etc.).
NTPI [NISO] {optional}	Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).
VEL [NGROUP] {optional}	Speeds for each group. Needed only for alpha calculations.
EBOUND [NGROUP+1] {optional}	Energy boundaries for each group.

ASCII text libraries may be entered in one of the four forms indicated by the IFIDO input. All four forms share the following features: Cross sections are entered in a table optionally preceded by a title line. A table consists of NGROUP rows of entries. Each row contains the cross sections for a single group and consists of IHM entries. The user specifies the positions in the row occupied by the total and selfscattering cross sections. Order within a row (e.g., for group g) is then as follows:

$$\ldots \sigma_{abs},\ \nu\sigma_f,\ \sigma_{total},\ \ldots \sigma_{g+2\rightarrow g},\ \sigma_{g+1\rightarrow g},\ \sigma_{g\rightarrow g},\ \sigma_{g-1\rightarrow g},\ \sigma_{g-2\rightarrow g},\ \text{etc.}$$

Notice that all terms in the scattering matrix are in positions relative to that of the self-scattering position and the rest of the cross sections are in positions relative to the position of the total cross section. The positions before the absorption cross section are frequently used for edit cross sections. For more detail, see "Ordering of Cross Sections Within a Cross-Section Table" on page 10-10.

Different Legendre orders are in different tables, which follow in order.

The user may order the group structure either by increasing energy or by decreasing energy. However, it is conventional and desirable for most problems to order it by decreasing energy, that is, group one is the highest energy. In that case, the scattering cross sections to the left of $\sigma_{g\rightarrow g}$ such as $\sigma_{g+1\rightarrow g}$ are upscattering terms and the terms to the right of $\sigma_{g\rightarrow g}$ are the downscattering terms.

In the Los Alamos format, the table is entered with a standard Fortran 6E12 format.

For greater precision in your input, use the 4E18 option.

In the fixed field FIDO format that ANISN[7] uses, entries are made in six twelve-column fields. Each twelve-column field is divided into three subfields, a two-column numeric field, a one-column character field, and a nine-column numeric field. See page 9-19 for details if you are not familiar with this input. The last field in each table must have the character T in the character position. No array identifier should be used. This format also restricts the usable input operators to T, *, R, -, +, and Z.

In the free field form, entries do not have to be in designated columns. Rather, the rules specified in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 apply. Each table in this form is also terminated with the character T. No array identifier (i.e., array name with appended equals sign) should be used.

Block-IV Details: Cross-Section Mixing

A short summary of the primary mixing arrays, MATLS and ASSIGN, is given here for quick reference. Normally, THESE TWO ARRAYS ARE REQUIRED and, in most problems, would be the only arrays in this block. Other mixing arrays are also briefly described.

There are actually several nested levels of mixing. Each level has the job of calculating values from expressions of the form: $\Sigma_g = \sum_{i=1}^{k} N_i \sigma_{i,g}$ for each group, g. The user's job is to input the N_i for all the k components of the mixture and to specify each component, i. Component i has the cross section, $\sigma_{i,g}$. In common usage, for the first level of mixing, $\sigma_{i,g}$ is the effective microscopic cross section and N_i is the atom density of isotope i, and Σ_g is then the macroscopic cross section of some material. In a higher level of mixing, these materials may be homogenized into a single material by using their volume fractions for the N_i. With several nested levels, the user has a great deal of flexibility in defining what Σ_g is for that level. A more complete discussion of mixing will be found in the chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1.

A discussion of cross section processing is outside the scope of this document, but it should be noted that the user needs to be aware of the processing that is inherent in the input library. For instance, for materials in which there are isotopes with cross-section resonances, self shielding of the cross sections for these isotopes may be important and this effect must have been considered in the preparation of the "effective" microscopic cross sections for these isotopes. Since the self shielding is dependent on the amounts and types of the other isotopes in the material, the "effective" cross section is strictly valid only for use in a mixture which has the same composition as was used in the self shielding calculation. If the user desires to use this same "effective" microscopic cross section in some other composition (mix) of material, it is up to the user to verify the accuracy of this approach.

Primary Mixing Arrays {Required}

Name	Description
MATLS[a] [-;MT]	Instructions for mixing "isotopes" or premixes into "materials." See details below.
ASSIGN[b] [-;NZONE]	Assignments of materials to geometric zones. See below.
PREMIX [-;-] {optional}	Instructions for mixing "isotopes" into premixes. See below.

a. The information entered in the MATLS array is written to the CCCC standard interface files NDXSRF and ZNATDN.

b. Information entered in the ASSIGN array is written to the code-dependent interface file ASGMAT.

In order to understand how cross sections are mixed and the resultant material placed in the problem, we first need a little conceptual information.

The key entities used in specifying the cross-section spatial distribution are coarse mesh, zone, isotope, and material.

The basic geometry of the problem is defined with the coarse meshes specified in Block-II. The geometric areas called zones are also defined there using the ZONES array; the ZONES array designates the zone number assigned to each coarse mesh.

Here in Block-IV, we mix cross sections and assign them to the zones created in Block-II. For the purposes of this discussion, the cross sections found on the input library belong, by definition, to "isotopes," no matter what their true nature. These "isotopes" may then be mixed to form materials, using the MATLS array. Materials are then assigned to zones using the ASSIGN array.

MATLS input array

The general form of a MATLS mix instruction is shown below:

$$\text{MATLS=}\ \ mat_1\ \ comp_1\ den_1,\ \ comp_2\ den_2,\ \text{...etc....}\ ;$$

where mat_1 is the desired character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional materials (i.e., mat_2, mat_3, and so on up to the required number, MT) are defined in subsequent strings. Each string may contain as many components as necessary (actual limit = 500). A component is usually an isotope from the library, but may also be a temporary material created by the PREMIX array (see below).

When the component is an isotope, the den_i is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 5-44).

Short form: MATLS= *ISOS*

This form specifies that there should be as many materials as isotopes and that isotope number 1 is to be used for material number 1, isotope number 2 is to be used for material number 2, and so on.

In the special case where there is only a single component in a material and its density is unity, the density entry may be omitted as in the first material below:

MATLS= mat_1 $comp_1$; mat_2 $comp_2$ den_2; ...etc.... ;

ASSIGN input array

The general form of the ASSIGN instruction is shown below:

ASSIGN= $zone_1$ mat_1 vol_1, mat_2 vol_2, ...etc.... ;

where $zone_1$ is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). mat_1, mat_2, and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" vol_1, vol_2, and so on. Additional zones (i.e., $zone_2$, $zone_3$, and so on up to the required number, NZONE) are defined in subsequent strings. Although it is highly recommended that you use character names, here it is convenient to use the numeral for the zone name because it is the same numeral entered in the ZONES array.

Short form: ASSIGN= *MATLS*

This form specifies that there are as many zones as there are materials, and that material number 1 is to be assigned to zone number 1, material number 2 to zone number 2, and so on.

NOTE: The short form ASSIGN=*MATLS* can not be used if you intend to use the ASGMOD input array described later in this section.

PREMIX input array

The PREMIX array forms temporary materials in a way exactly analogous to the way that permanent materials are formed in the MATLS array. The difference in treatment is that the temporary materials created by PREMIX exist only long enough to complete the mixing; they are not available for assignment to geometric zones, nor are they available for use in material edits.

The general form of a PREMIX mix instruction is shown below:

PREMIX= $tmat_1$ $comp_1$ den_1, $comp_2$ den_2, ...etc.... ;

where $tmat_1$ is the character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional temporary materials (i.e., $tmat_2$, $tmat_3$, and so on) may be defined in subsequent strings. A component may be either an isotope from the library or another temporary material created by PREMIX.

The PREMIX array is useful for organizing the mixing input. For instance, it is frequently useful to mix the cross sections for a molecule of water and then in subsequent mix instructions, to input the molecular density of water as opposed to entering the atom density for both hydrogen and oxygen. Other examples are to form average cross sections for an element composed of many isotopes, or to form full density materials and then in later mix instructions to put in the volume fraction of the full density material.

Character Names vs. Numeric Names

In the foregoing discussion, isotopes, materials, and zones were identified by their character names. Optionally, they may be referred to by their ordinal number. Thus, 2 for an isotope name would call for the second isotope on the library. However, this practice is NOT recommended.

THE CHARACTER NAME FORM IS HIGHLY RECOMMENDED. It provides the most straightforward, most readable form. If the character name form is used, the naming input arrays in the following table are not needed.

Using the character name form in one array and the numeric name form in another array is particularly discouraged. However, should one wish to use the numeric form in the MATLS and/or ASSIGN arrays, and then subsequently associate character names with the ordinal numbers, one can use the naming arrays in the following table to do so. This situation could arise if, for some reason, one wanted to use material numbers in the MATLS array, but use character material names in the ASSIGN array.

When the library is of the MENDF form, the character names that must be used for the isotope names are discussed in "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.

Concentration Search

The concentration search has not been implemented in TWOHEX.

Miscellaneous Mixing Input {Optional}

Name	Comments
MATNAM [MT]	Character material names for Materials. Used only if the mat_1 name used in the MATLS array was integer. First entry in MATNAM array is the desired character name for Material number 1, second entry is the desired character name for Material number 2, etc.
ZONNAM [NZONE]	Character zone names for Zones. Used only if the zone name entry in the ASSIGN or ASGMOD array was integer. First entry in the ZONNAM array is the desired character name for Zone number 1, second entry is the desired character name for Zone number 2, etc.
MATSPEC [≤MT]	Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions. Allowable entries are the words: *ATDENS* (default) atomic densities *ATFRAC*[a] atomic fractions *WTFRAC* weight fractions Can be input as a vector with up to MT entries (one for each Material) [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13.] If less than MT entries are made, the last entry will be used to fill out the array to a length of MT.
ATWT [≤2*NISO] {required[b]}	Atomic weights of the isotopes. If using MATSPEC=ATFRAC or WTFRAC, atomic weights must be available to the code. Entries for the ATWT array are made in pairs, as follows: ATWT= iso_1 $atwt_1$ iso_2 $atwt_2$ .... where iso_n is the isotope name (identifier) for isotope n on the cross-section library and $atwt_n$ is that isotope's atomic weight. [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13].

a. ATFRAC and WTFRAC cannot be used with PREMIX.

b. Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.

Block-V Details: Solver Input

Desired Calculation {Required}

Name	Comments
IEVT	Calculation type: Enter one of the following values: Value — Description *0* — source *1* — k_{eff}[a]
ISCT	Legendre order of scattering.
ITH	*0/1* = direct/adjoint calculation.

a. For a k_{eff} calculation, an inner iteration sequence for a given group is terminated when the maximum scalar flux error is reduced to .04 times its value for the first inner iteration in the sequence. The outer iteration is terminated when the ratio of fission sources from outer to outer is within EPSI of unity and the maximum pointwise fission source error is less than 5 times EPSI. A source plus fission calculation must meet the same convergence criteria as a k_{eff} calculation. In an inhomogeneous problem the maximum scalar flux error must be less than EPSI for the problem to terminate.

Iteration Controls {Required}

Name	Comments
EPSI	Convergence precision (default=0.0001).
OITM	Maximum no. of outer iterations (default=20).
ITLIM	Number of seconds time limit (default=unlimited).

Output Controls {Optional}

Name	Comments
FLUXP	Final flux print. *0/1/2* = no/isotropic/all moments.
XSECTP	Cross-section print. *0/1/2* = no/principal/all.
FISSRP	Fission rate print. *0/1*= no/yes.
SOURCP	Source print. *0/1/2/3* = no/unnormalized/normalized/both.

Miscellaneous Solver Input {Optional}

Name	Comments
NORM	Normalize the fission source rate to this value when IEVT≥1 or normalize the inhomogeneous source rate to this value when IEVT<1. NORM=0 means no normalization. (Integral of source rate over all angle, space, and energy = NORM, except for k_{eff} problems where the integral is equal to NORM*k_{eff}.) Any fluxes printed here (i.e., caused by setting FLUXP nonzero) will be normalized consistently with this source rate.
BHGT	Buckling height. (in cm. if macroscopic cross section is in cm^{-1}.) (default=0.0 which is treated as infinity.)
CHI [NGROUP;M]	Fission fraction born into each group.[a] Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.
DENX [IT][b] and/or	Density factor to use for each x-mesh column (default=1). Applied to the zone macroscopic cross sections at each mesh interval.
DENY [JT]	Density factor to use for each y-mesh band (default=1).

a. This input will override any previous CHI from earlier blocks or from any cross-section library which contains CHI.

b. The density factor DEN(i,j), at mesh interval (i,j) is computed as follows:

$$DEN(i,j) = DENX(i)*DENY(j)$$

Flux Guess From a File {Optional}

Name	Comments
INFLUX	Read the initial flux guess from a file.[a] *0/1* = no/yes. If ITH=0, read the initial flux guess from the RTFLUX file. If ITH=1, read the initial flux guess from the ATFLUX file.

a. There is presently no text input flux guess available for TWOHEX.

Quadrature Details {Required}

Name	Description
IQUAD	Type of angular quadrature. Enter one of the following: Value — Description *0* — Default[a] *1* — Constant[b] *2* — Triangular[c] *3* — Rectangular[d]

a. The default set uses the constant arrangement for ISN=4, and the triangular arrangement for all higher ISN values. Each of these arrangements is described below where we define xi= z direction cosine, mu= x direction cosine, and eta= y direction cosine. Examples are shown in Figure 5.2. The number of xi levels is equal to ISN/2, the S_n order divided by two.

b. The constant quadrature arrangement has one mu,eta point on each xi level in each sextant of the hemisphere.

c. The triangular arrangement for ISN/2 levels has, per sextant, one mu,eta point on the highest (numerically largest) xi level, two mu,eta points on the next highest xi level, up to ISN/2 mu,eta points on the lowest (numerically smallest) xi level.

d. The rectangular arrangement for ISN/2 xi levels has, per sextant, ISN/2 mu,eta points on each xi level.

Note: The S_2 quadrature set is independent of IQUAD value.

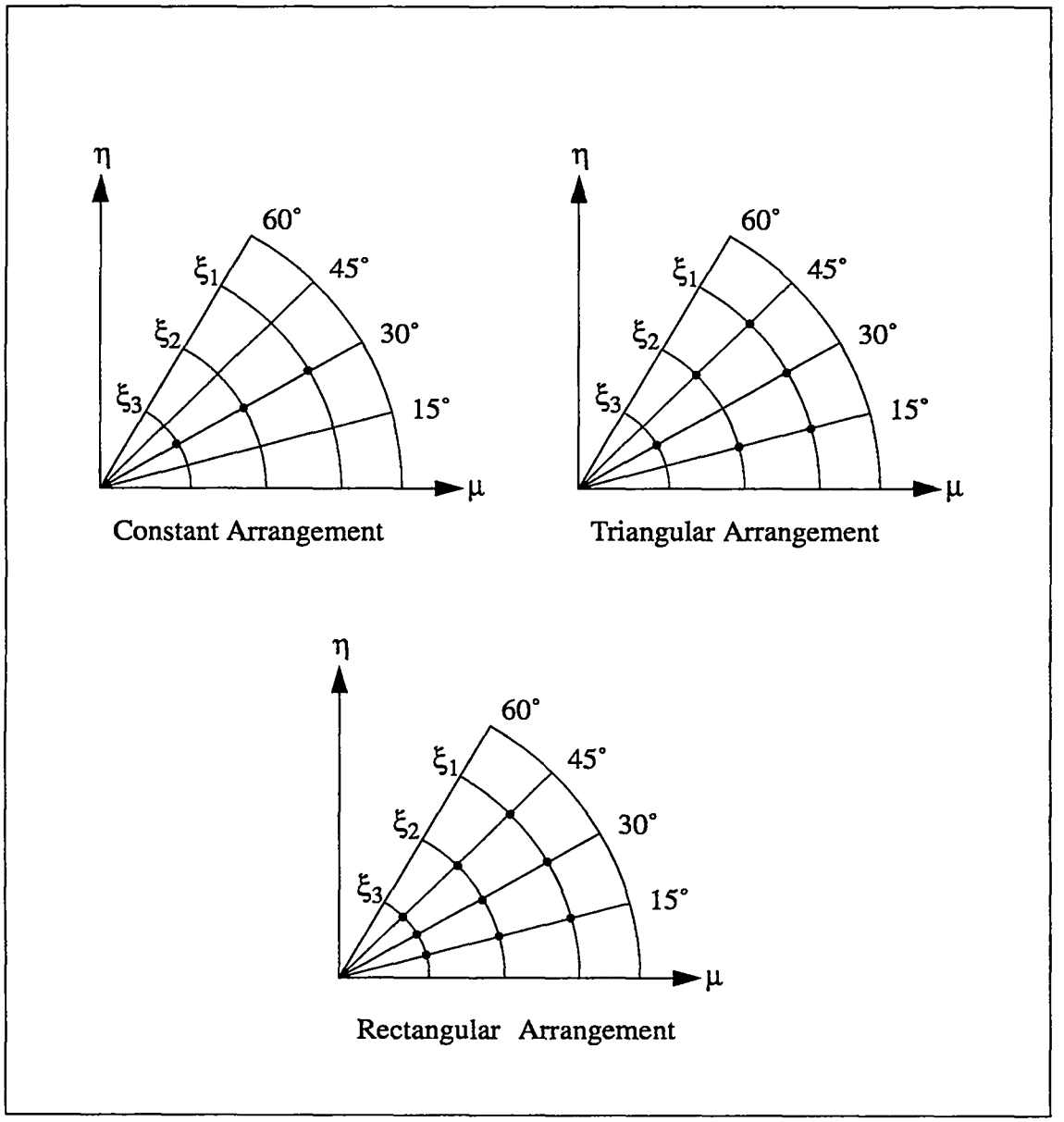


Figure 5.2 Possible S_6 Quadrature Arrangements

Figure 5.2 illustrates the quadrature point arrangement in a sextant for each of the IQUAD values for the specific case where ISN=6.

Volumetric Source Options {Optional}

Name	Comments
INSORS	Read source from interface file FIXSRC. *0/1* = no/yes.
----- For a text-input source, choose one of the following options:	
Option 1:	
SOURCE [NGROUP; NMQ]	Source spectrum for each of NMQ[a] moments. (Spatial distribution is assumed to be flat with value unity)
Option 2:	(input both arrays)
SOURCX [IT;NMQ][b]	x spatial distribution for each moment.
SOURCY [JT;NMQ]	y spatial distribution for each moment.
(Spectrum is assumed to be flat with value unity)	
Option 3:	(input all three arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCX [IT;NMQ]	x spatial distribution for each moment.
SOURCY [JT;NMQ]	y spatial distribution for each moment.
Option 4:	
SOURCF [IT;JT*NGROUP*NMQ]	Spatial distribution for each row, group, and moment.
Option 5:	(input both arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCF [IT; JT*NMQ]	Spatial distribution for each row and moment.

a. NMQ is not an input value but is computed from the number of strings read. NMQ must correspond exactly to the number of moments in a P_n expansion of the source. The number of moments is (n+1)(n+2)/2. n must be less than or equal to ISCT. See page 12-24 for more details.

b. Only in option 4 is the complete pointwise source array, SOURCF(i,j,g,m), given. In all other cases, it must be formed from the lower dimension arrays that are input. That calculation is done by forming the product of those arrays. Thus, in option 3, where the source spectrum, SOURCE(g,m), and the spatial distributions SOURCX(i,m), SOURCY(j,m), are given (for moment m), the full source at mesh point (i,j) in group g for moment m is calculated as follows:

$$SOURCF(i,j,g,m) = SOURCE(g,m)*SOURCX(i,m)*SOURCY(j,m)$$

Block-VI Details: Edit Input

Edit Spatial Specifications {Required[a]}

Name	Comments
PTED	Do edits by mesh interval. *0/1* = no/yes.
ZNED	Do edits by zone. *0/1* = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below).
POINTS[≤IT*JT] {optional}	Mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points)
EDZONE [IT;JT] {optional}	Edit zone number for each mesh interval. USED ONLY IF ZNED=1. (default= SOLVER coarse mesh interval numbers, see ZONES array, Block-II on page 5-33).

a. Either PTED or ZNED or both must be unity in order to produce reaction rate edits.

Reaction Rates from Cross Sections[a] {Optional[b]}

Name	Comments
EDXS [≤NEDT] {required[c]}	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 the character name of the cross-section type. See the table "Edit Cross-Section Types by Position and Name" on page 5-52 or "MENDF Library Edit Cross Sections" on page 5-58 for the available names. NEDT is the total number of edit cross-section types available from the input cross-section library. (default = all shown in the table) Note: The cross-section types specified in this array apply to any or all of the following edit forms: RESDNT, EDISOS, EDCONS, EDMATS.
RESDNT	Do edits using the resident macroscopic cross section at each point. *0/1* = no/yes.
EDISOS [≤NISO]	Character names of the isotopes to be used in forming Isotopic reaction rates. The ordinal number may alternatively be used but is not recommended. (default = none).
EDCONS [≤NISO]	Character names of the isotopes to be used in forming resident Constituent (partial macroscopic) reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDMATS [≤MT]	Character names of materials to be used in forming Material (macroscopic) reaction rates. The ordinal number may alternately be used, but is not recommended. (default = none).
XDF[d] [IT] YDF [JT]	Mesh density factors for the x and y directions, respectively. The density factor is used to multiply resident Constituent (see EDCONS), macroscopic (see MACRO), and resident macroscopic (see RESDNT) reaction rates only. (default= all values unity).

a. See chapter "RUNNING THE EDIT MODULE" starting on page 8-1 for further discussion.

b. But either something in this grouping or in the "Reaction Rates from User Response Functions" grouping must be input in order to produce reaction rate edits.

c. You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.

d. If density factors were used in SOLVER to modify the cross sections at each mesh interval, the same density factors must be provided here in the XDF and/or YDF arrays as well. The density factor at mesh interval (i,j) is computed as:

XDF(i)*YDF(j)

Edit Cross-Section Types by Position and Name

CROSS-SECTION INPUT VIA ISOTXS or GRUPXS			CROSS-SECTION INPUT VIA ASCII TEXT		
Type	EDIT Position	Name[a]	Type	EDIT Position	Name
chi	1	CHI.....	not used	1	CHI.....
nu-fission	2	NUSIGF..	nu-fission	2	NUSIGF..
total	3	TOTAL...	total	3	TOTAL...
absorption	4	ABS.....	absorption	4	ABS.....
(n,p)	5	N-PROT..	1[b]	5	EDIT1...[c]
(n,d)	6	N-DEUT..	2	6	EDIT2...
(n,t)	7	N-TRIT..	3	7	EDIT3...
(n,alpha)	8	N-ALPH..	.	.	.
(n,2n)	9	N-2N....	.	.	.
(n,gamma)	10	N-GAMM..	.	.	.
fission	11	N-FISS..	N=IHT-3	4+N	EDITN...
transport	12	TRNSPT..			

a. Names are eight characters. A period within a name in this table denotes a blank.

b. Denotes position in the cross-section table. All cross sections in 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 EDNAME array of Block-III.

Reaction Rates from User Response Functions {Optional[a]}

Name	Comments
RSFE [NGROUP;M] {required}	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 with NGROUP entries beginning with group 1.
RSFX [IT;M][b]	Response function x distribution for M functions.
RSFY [JT;M]	Response function y distribution for M functions.
{optional}	The above data are entered as M strings of IT or JT entries beginning with mesh point 1. (default=1.0)
RSFNAM [M]	Character names for the user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM)

a. But either something in this grouping or in the "Reaction Rates from Cross Sections" grouping must be input in order to produce reaction rate edits.

b. The M-th response function at space point (i,j) and energy group g is computed as:

RSFX(i,m)*RSFY(j,m)*RSFE(g,m)

Energy Group Collapse Specifications {Optional}

Name	Comments
ICOLL [NBG]	Edit energy group collapsing option: Number of SOLVER energy groups in each EDIT broad group. The NBG entries must sum to NGROUP. (default = 1 energy group per EDIT broad group).
IGRPED	Print option on energy groups. Enter one of the following values: Value — Description *0* — Print energy group totals only *1* — Print broad groups only *2* — Print broad groups only (same as 1) *3* — Print both broad groups and totals

Reaction Rate Summing {Optional}

Name	Comments
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 "Edit Cross-Section Types by Position and Name" on page 5-52). Each of 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. (for more detail, see "Response Function Summing Options" on page 8-13).
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 page 8-13 for more detail.

Mass Inventories {Optional}

Name	Comments
MASSED	Calculate and print mass inventories by zone. *0/1/2/3* = none/ solver zones/edit zones/both (default=1). This option is active only if atomic weights are present. See ATWT on page 5-44.

Power Normalization {Optional}

Name	Comments
POWER {required}	Normalize to POWER megawatts.[a] All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be normalized. Fluxes are normally not printed here in the EDIT module, although they may be extracted by using a unit response function. Any such fluxes will also be normalized to POWER. Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 5-45).
MEVPER {required}	MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power. For the power calculation, TWOHEX needs to know which cross section is the fission cross section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or GRUPXS library that designation is automatically provided (See "Edit Cross-Section Types by Position and Name" on page 5-52). But if one uses an ASCII text library, either ODNINP or XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 5-38).

a. Note that this normalization is meaningless if you are using the results of an adjoint run.

Miscellaneous Edit Items {Optional}

Name	Comments
RZFLUX	Write the CCCC standard zone[a] flux file RZFLUX or AZFLUX. *0/1* = no/yes.
RZMFLX	Write the code-dependent zone[b] flux moments file RZMFLX or AZMFLUX. *0/1* = no/yes.
EDOUTF[c]	ASCII output files control. Enter one of the following values:

Value	Description
-3	Write both EDTOGX (without scalar fluxes) and EDTOUT files.
-2	Write EDTOGX file (without scalar fluxes).
0	Write neither file. (default)
1	Write EDTOUT file.
2	Write EDTOGX file (with scalar fluxes).
3	Write both EDTOGX (with scalar fluxes) and EDTOUT files.

BYVOLP	Printed point reaction rates will have been multiplied by the mesh volume. *0/1* = no/yes.
AJED[d]	Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. *0/1* = regular/adjoint.
FLUXONE	Flux override. *0/1* = no/yes. Replaces all the input fluxes by unity. Useful for seeing the cross sections used in cross-section edits. WARNING! Meaningful reaction rates cannot be obtained when this switch is on.

a. RZFLUX and AXFLUX are organized by solver zones.

b. RZMFLX and AZMFLUX are organized by solver zones.

c. See "ASCII File Output Capabilities (the EDOUTF Parameter)" on page 8-15.

d. See "Adjoint Edits" on page 8-15.

MENDF Library Edit Cross Sections

Reaction Type	Name	Description
χ	CHI	fission spectrum
$\nu\sigma_f$	NUSIGF	effective nu-sigma-fission
σ_t	TOTAL	Total cross section
σ_a	ABS	absorption[a]
(n,n)	MEND1	elastic scattering
(n,n´)	MEND2	inelastic scattering
(n,2n)	MEND3	n,2n scattering
(n,3n)	MEND4	n,3n scattering
(n,γ)	MEND5	gamma production
(n,α)	MEND6	alpha production
(n,p)	MEND7	proton production
(n,f)	MEND8	direct fission
(n,n´)f	MEND9	second-chance fission
(n,2n)f	MEND10	third-chance fission
(n,F)	N-FISS	[(n,F) = (n,f) + (n,n´)f + (n,2n)f]
χ_p	MEND12	prompt fission spectrum (only for fissionable materials)
χ_t	MEND13	total fission spectrum (only for fissionable materials)

a. σ_a for group g is defined as $\sigma_a = \sigma_t - \sum_{g'} \sigma_{g \to g'}$.

When using the Los Alamos MENDF5 cross-section library with the codes, there are numerous edit cross sections available for use in the Edit Module. Since these come from the MENDF file, they are called upon with special character names in the Edit Module as part of the EDXS input.

These names are defined in the table above.

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. D. R. Ferguson and K. L. Derstine, "Optimized Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes," Nucl. Sci. Eng. **64**, 593 (1977).
3. K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," Argonne National Laboratory report ANL-82-64 (April 1984).
4. W. F. Walters, "The TLC Scheme for Numerical Solution of the Transport Equation on Equilateral Triangular Meshes," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 151-165.
5. W. F. Walters, F. W. Brinkley, and D. R. Marr, "User's Guide for TWOHEX: A Code Package for Two-Dimensional, Neutral-Particle transport in Equilateral Triangular Meshes," Los Alamos National Laboratory report LA-10258-M (October 1984).
6. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).
7. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693 (March 1967).

APPENDIX A: SAMPLE INPUT

Sample Problem: Standard k_{eff} Calculation.

Our sample problem is a two group calculation of the eigenvalue of a sodium-cooled fast reactor. The reactor contains two zones, a core zone composed of two rows of fuel subassemblies surrounding a central fuel subassembly and a reflector zone consisting of one row of reflector assemblies surrounding the core zone. This arrangement is shown in Figure 5.3. The reactor is surrounded by void and with the dentate periphery, it is possible for some neutrons to exit a face and reenter another, although most would be lost.

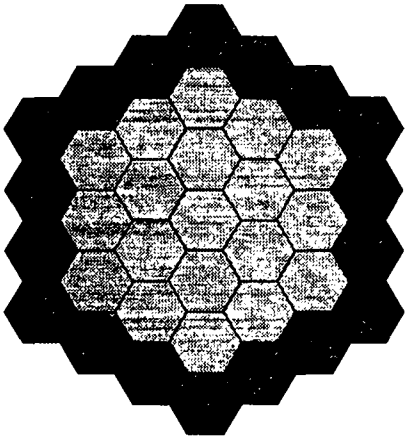


Figure 5.3 Core Map of the Sample Problem

In modeling this reactor in TWOHEX, we take advantage of the symmetry and only model a sixty degree sector extending from the core center upwards to the right as shown in Figure 5.4. Each subassembly is modeled with 6 triangles. In the sixty-degree sector option, there are implicit rotational boundary conditions, that is, flux leaving the bottom face enters the slanted left face. Leakage in the axial dimension is approximated with a transverse buckling.

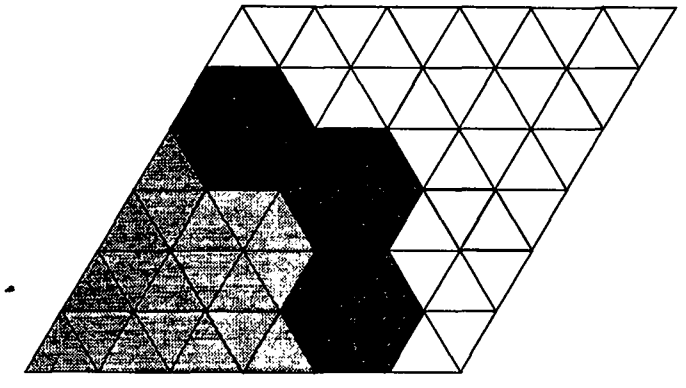


Figure 5.4 Mesh Model for the Sample Problem

A note of warning about modeling is in order here. This sample problem utilizes a sixth core model for which there are "vacuum" boundary conditions on the top and right faces of the outer periphery of Figure 5.4. These boundaries are really non-reentrant boundaries (i.e., the flux passing outward through these boundaries is lost and does not reenter a boundary anywhere else). Note that the sample model contains extra void triangles around the exterior of the reflector. This is to provide a streaming path within the model for flux leaving the reflector that truly can reenter another portion of the reflector. Removal of these extra void triangles will cause the eigenvalue and fluxes to be slightly lower.

Sample Problem: Output Description

Selected items in the output listing are described here. We focus on items particular to a two-dimensional calculation. For a more thorough description of output items common to both one- and two-dimensional calculations, the reader is referred to Appendix A in the chapter "ONEDANT USER'S GUIDE".

The first item provided in the output is a listing of the input lines. This is shown on page 5-64. Note the use of comments (using the slash, /) to organize and describe the input.

After the Block-I input describing the problem as a whole and the Block-II input describing the geometry and zone assignments by mesh, we describe the cross-section library in Block-III.

P_0 cross sections for each isotope are entered in the input stream after Block-III. These isotopes are subsequently mixed in Block-IV to form the materials STEEL, FUEL, and SODIUM. These materials are then assigned with appropriate volume fractions to the CORE and REFLECT zones which correspond to the numbers 1 and 2 in the ZONES array of Block-II.

In Block-V we specify that this is a k-eigenvalue calculation (ievt=1) and that we desire a calculation using only isotropic scattering (isct=0). We approximate the axial leakage with a transverse buckling consistent with a height of ninety centimeters. The fission source is normalized to 1.0. We desire no printout of the fluxes, the cross sections, or the fission rate. The fission fractions in each of the two groups is, respectively, 0.6 and 0.4 for the first zone, and 0.7 and 0.3 for the second zone.

In the edit input, the code is asked to calculate reaction rate totals for each mesh interval (pted=1). The reaction rates desired are the default ones for cross sections, that is, CHI, NUSIGF, TOTAL, ABS, and EDIT1 for the resident material.

We then start a series of sections where the code prints out the Block-I through Block-IV input as it understands it. This is being done as the input module processes each block of input. There is only a notation as the input module processes the Block-V input for the Solver module and the Block-VI input for the Edit module, for when control is passed to those modules, the input will be reiterated there.

After the notation that we are at the end of the input module, we start the Solver module output on page 5-68. Note that in the first table, the code has "defaulted" the boundary conditions. These are built in to each of the geometry options. The "sixth" option dictates rotational boundary conditions on the left and bottom faces along with vacuum on the top and right.

As far as the rest of the output shown, most of the items are also self explanatory and give the same information as a ONEDANT problem except for the two-dimensional differences. However, we would like to focus a little more attention on two items that serve as diagnostics and goodness of solution verification for the problem run: the iteration monitor and the balance table.

In considering the iteration monitor (page 5-70), we recall that for eigenvalue problems we do source iteration for the inner or within-group scattering source and outer iterations for the fission source. This is reflected in the monitor which is arranged in rows for each outer. The inner convergence is not shown until the fission source has converged to near the input convergence criterion. Each column of the monitor gives respectively, the current CPU time, the transport outer counter, the number of transport inners for this outer, the eigenvalue estimate at this outer, the precision of the eigenvalue (change from the previous transport outer), the estimated dominance ratio and the dominance ratio actually used. Note that for outer 7 the source has sufficiently converged where now the inner iterations on the groupwise scalar flux can now be meaningfully shown. After the completion of outer iteration 10, the whole problem has been converged. From the first column, we see that this was done in 0.4 CPU seconds on the Cray YMP.

In the edit input, the code is asked to give reaction rate totals for each triangle for the resident material. The reaction rates desired are the default ones, that is, CHI, NUSIGF, TOTAL, ABS, and EDIT1 (EDIT1 is the default name for the first position in the ASCII text library). This reiteration of the input is followed by the pointwise reaction rate tables.

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*          generalized input module run on 03/24/95 with solver version 01-04-95beta--- release 2.6mx    machine e
*
*****************************************************************************************************************************
*                  ***************************************************************************************
*                  *                                                                                     *
*                  *                    ...listing of cards in the input stream...                       *
*                  *                                                                                     *
*                  *    1.     2     0     0                                                             *
*                  *    2. SAMPLE PROBLEM FOR TWOHEX USER'S GUIDE                                        *
*                  *    3. STANDARD K CALCULATION,  ALL INPUT BY MEANS OF ASCII TEXT                     *
*                  *    4. /   GEOMETRY         - SIXTH CORE WITH PERPENDICULAR BUCKLING                 *
*                  *    5. /   CROSS SECTIONS - 2 GROUP, ISOTROPIC SCATTER                               *
*                  *    6. /                    ISOTOPE DATA IN TEXT,  LOS ALAMOS (DIF) FORMAT           *
*                  *    7. /   MIXING           - ISOTOPES MIXED TO MAKE MATERIALS NAMED STEEL,          *
*                  *    8. /                      FUEL, AND SODIUM                                       *
*                  *    9. /                    - MATERIALS ASSIGNED TO MAKE ZONES NAMED CORE            *
*                  *   10. /                      AND REFLECTOR                                          *
*                  *   11. /   SOLVER          - ASCII TEXT INPUT SUPPLIED                              *
*                  *   12. /   EDITS           - POINT EDITS FOR RESIDENT MATERIALS                     *
*                  *   13. /                                                                            *
*                  *   14. / ------------------- BLOCK I -------------------                            *
*                  *   15.   igeom=hex, ngroup=2, isn=4, niso=7, mt=3, nzone=2, it=12, jt=6, t          *
*                  *   16. /                                                                            *
*                  *   17. / ------------------- BLOCK II (GEOMETRY) -------------------                 *
*                  *   18.     domain=sixth,   height=10.0,                                             *
*                  *   19.   zones=      /                                                              *
*                  *   20.   7r1 3r2 f0;  6r1 3r2 f0;  5r1 3r2 f0; / ZONE NOS. FOR BANDS 1,2,3         *
*                  *   21.     1 6r2 f0;       3r2 f0;  f0;        t / ZONE NOS. FOR BANDS 4,5,6         *
*                  *   22. /                                                                            *
*                  *   23. / ------------------- BLOCK III (CROSS SECTIONS) -------------------         *
*                  *   24.     lib= cdninp                                                              *
*                  *   25.     maxord=0  ihm=6  iht=4  ihs=5  ifido=0  ititl=1                          *
*                  *   26.     names= "O-16"  "NA-23"  FE  CR  NI  "PU-239"  "U-238"                    *
*                  *   27. /                                                                            *
*                  *   28. / ****** SINCE LIB=CDNINP, THE CROSS SECTION LIBRARY IN ASCII TEXT           *
*                  *   29. /  WILL BEGIN IMMEDIATELY FOLLOWING THE BLOCK III TERMINAL "T".              *
*                  *   30. /   NOTE THAT A TITLE LINE PRECEDES EACH CROSS-SECTION                       *
*                  *   31. /   BLOCK (SINCE ITITL=1). ******                                           *
*                  *   32. /                                                                            *
*                  *   33.   t                                                                          *
*                  *   34. OXYGEN-16  (O-16)     SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   35.         0.000       0.010       0.000       2.000       1.600       0.000 O16/1 *
*                  *   36.         0.000       0.000       0.000       3.600       3.600       0.390 O16/2 *
*                  *   37. SODIUM      (NA-23)   SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   38.         0.000       0.002       0.000       1.900       1.500       0.000 NA23/1*
*                  *   39.         0.000       0.005       0.000       4.000       3.995       0.398 NA23/2*
*                  *   40. IRON        (FE)      SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   41.         0.000       0.008       0.000       2.100       1.700       0.000 FE/1  *
*                  *   42.         0.000       0.010       0.000       4.500       4.490       0.392 FE/2  *
*                  *   43. CHROMIUM    (CR)      SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   44.         0.000       0.013       0.000       2.450       2.150       0.000 CR/1  *
*                  *   45.         0.000       0.020       0.000       5.000       4.980       0.287 CR/2  *
*                  *   46. NICKEL      (NI)      SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   47.         0.000       0.080       0.000       2.400       2.000       0.000 NI/1  *
*                  *   48.         0.000       0.030       0.000       8.000       7.970       0.320 NI/2  *
*                  *   49. PLUTONIUM   (PU-239)  SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   50.         1.900       1.950       6.270       4.800       2.000       0.000 PU239/1*
*                  *   51.         1.600       2.500       4.800       12.00       9.500       0.850 PU239/2*
*                  *   52. URANIUM     (U-238)   SAMPLE 2 GROUP LWFR CROSS SECTIONS                     *
*                  *   53.         0.300       0.400       0.900       4.700       3.000       0.000 U238/1*
*                  *   54.         0.000       0.500       0.000       13.00       12.50       1.300 U238/2*
*                  *   55. /                                                                            *
*                  *   56. / ******* END OF CROSS-SECTION DATA ******                                   *
*                  *   57. / **** NOTE THAT THERE IS NO TERMINAL "T" SINCE THE CROSS SECTIONS ARE       *
*                  *   58. /       IN LOS ALAMOS (DIF) FORMAT (IFIDO=0) ****                           *
*                  *   59. /                                                                            *
*                  *   60. / ------------------- BLOCK IV (MIXING) -------------------                   *
*                  *   61.     matls= STEEL, FE .05, CR .016,  NI 0.01;                                 *
*                  *   62.            FUEL  "PU-239" .0081,  "U-238" .0125    "O-16" .0412;             *
*                  *   63.            SODIUM  "NA-23" .025                                              *
*                  *   64.     assign= CORE    FUEL  .35, SODIUM .4, STEEL .25;                         *
*                  *   65.             REFLEC  SODIUM .7, STEEL  .3              t                      *
*                  *   66. /                                                                            *
*                  *   67. / ------------------- BLOCK V (SOLVER) -------------------                   *
*                  *   68.     ievt=1   isct=0    bhgt=90.    norm=1.0                                  *
*                  *   69.     fluxp=0  xsectp=0  fissrp=0    chi=0.6,0.4; 0.7, 0.3     t               *
*                  *   70. /                                                                            *
*                  *   71. / ------------------- BLOCK VI (EDITS) -------------------                   *
*                  *   72.     pted=1, resdnt=1, t /  POINT EDIT FOR RESIDENT MATERIALS                 *
*                  *                                                                                     *
*                  ***************************************************************************************
*****************************************************************************************************************************
```

```
**********************************************************************************************************************************
*
*                                                  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 FOR TWOHEX USER'S GUIDE                                  *
*                          * STANDARD K CALCULATION,  ALL INPUT BY MEANS OF ASCII TEXT               *
*                          *                                                                         *
*                          ***************************************************************************
************************
*key end   block i read*
************************
**********************************************************************************************************************************

**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                        ...block i - controls and dimensions...
*
**********************************************************************************************************************************
*
*
*                                             ...dimensions ...
*
*                           9  igeom   6/7/9/11/106/107  x-y/r-z/triangles/r-theta/gq x-y/gq r-z
*                           2  ngroup  number of energy groups
*                           4  isn     angular quadrature order
*                           7  niso    number of input isotopes (from isotxs, grupxs, or cards)
*                           3  mt      number of permanent materials
*                           2  nzone   number of zones
*                          12  it      number of fine mesh intervals
*                           6  jt      number of fine mesh y intervals
*
*                                                ...storage...
*
*                                               maxlcm=  140000
*                                               maxscm=   40000
*
**********************************************************************************************************************************
******************************
*key end   block ii read-geom*
******************************
*
*                         9  igeom    1/2/3/6/7/8/9/11/14/106/107
*
*                     sixth  domain   calculational domain
*
*              1.000000E+01  height   triangle height
******************************
*key end   block iii read-xs *
******************************

**********************************************************************************************************************************
```

```
*********************************************************************************************************************************
*                                                                                                                               *
*                                        ...block iii - cross section library...                                                *
*                                                                                                                               *
*********************************************************************************************************************************
*
*                                              ...library source...
*                                                 lib=cdnlnp
*
*                                        ...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   -1/0/1/2 - dtf(4e18.0)/dtf/fixed fido/free fido library
*                      1  iriti   0/1 - no/yes there is a title card before each table
*                      0  i2lpl   0/1 - no/yes library higher order scattering contains 21+1 factor
*                      0  savbxs  0/1 - no/yes save binary xslib file (filename=bxslib)
*                      1  kwikrd  0/1 - full fido read/quick fido read (default=quick)
*
*
*                                              ...energy structure...
*
*   group     chi          vel        lower bound   upper bound   group     chi          vel        lower bound   upper bound
*     1   0.00000E+00  0.00000E+00   0.00000E+00   0.00000E+00       2   0.00000E+00  0.00000E+00   0.00000E+00   0.00000E+00
*
*                                    last neutron group(lng) is number   2
*
*
*                      0  balxs   -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance
*
*
*                                        ...edit position names...
*
*                                                                 card
*                                        position  edname  position
*                                        --------  ------ --------
*                                            1     chi
*                                            2     nusigf      3
*                                            3     total       4
*                                            4     abs         2
*                                            5     edit1       1
*
*********************************
*key start card libe read    *
*********************************
*
*                                        ...header cards from the card library...
*
*                                isotope isotope
*                                number  name   order          header card
*                                ------  -----  -----  ------------------------------
*                                   1.   O-16    p0    -  OXYGEN-16  (O-16)    SAMPLE 2 GROUP LMFBR CROSS SECTIONS
*                                   2.   NA-23   p0    -  SODIUM     (NA-23)   SAMPLE 2 GROUP LMFBR CROSS SECTIONS
*                                   3.   FE      p0    -  IRON       (FE)      SAMPLE 2 GROUP LMFBR CROSS SECTIONS
*                                   4.   CR      p0    -  CHROMIUM   (CR)      SAMPLE 2 GROUP LMFBR CROSS SECTIONS
*                                   5.   NI      p0    -  NICKEL     (NI)      SAMPLE 2 GROUP LMFBR CROSS SECTIONS
*                                   6.   PU-239  p0    -  PLUTONIUM  (PU-239)  SAMPLE 2 GROUP LMFBR CROSS SECTIONS
*                                   7.   U-238   p0    -  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   8.10000E-03, U-238    1.25000E-02, O-16     4.12000E-02,
*       3. SODIUM   NA-23    2.50000E-02,
*
*     assign
*     ------
*
*       1. CORE     FUEL     3.50000E-01, SODIUM   4.00000E-01, STEEL    2.50000E-01,
*       2. REFLEC   SODIUM   7.00000E-01, STEEL    3.00000E-01,
************************
*key start mix card xs *
************************
************************
*key end   mix card xs *
************************
*****************************************************************************************************************************************
*****************************
*key end   block v read-solvr*
*****************************
*****************************
*key end   block vi read-edit*
*****************************
************************
*key end   input module*
************************
*
*
*****************************************************************************************************************************************
```

```
*****************************************************************************************************************************************

                this twohex problem run on 03/24/95 with solver version 12-06-94*product release 2.6mx    machine e
*SAMPLE PROBLEM FOR TWOHEX USER'S GUIDE
*STANDARD K CALCULATION,  ALL INPUT BY MEANS OF ASCII TEXT
*
*****************************************************************************************************************************************
*
*                                        ...block v -- solver input...
*
*****************************************************************************************************************************************
*          raw     as
*         input  defaulted
*         -----  --------
*
*                                        ...required input(array name = solin)...
*
*         1         1       ievt   0/1 - type of calculation
*                                      0 inhomogeneous source
*         0         0       isct   legendre order of scattering
*         0         0       ith    0/1 - direct/adjoint - mode of calculation (default=direct)
*         0         4       ibl    2/4 - left boundary condition
*                                      vacuum/rotational
*         0         2       ibr    2 - right boundary condition
*                                      vacuum
*         0         2       ibt    2 - top boundary condition
*                                      vacuum
*         0         4       ibb    2/4 - bottom boundary condition
*                                      vacuum/rotational
*
*
*                                        ...convergence controls(array name = iter)...
*
*      0.000E+00 1.000E-04  epsi   inner iteration convergence criterion (default=0.0001)
*         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
*         -----  --------
*
*
*                                        ...miscellaneous parameters(array name = misc)...
*
*      9.000E+01 9.000E+01  hbgt   buckling height
*      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         1       iquad  0/1/2/3 - quadrature point arrangement
*                                           default/constant/triangular/rectangular
*
*
*                                        ...output controls(array name = solout)...
*
*                     0  fluxp  0/1/2   none/isotropic/all moments - flux print
*                     0  xsectp 0/1/2   none/principal/all - macroscopic cross section print
*                     0  fissrp 0/1     no/yes - print final fission source rate
*                     0  sourcp 0/1/2/3 no/as read/normalized/both - print inhomogeneous source
*                     0  angp   0/1     no/yes - print angular fluxes (unimplemented)
*                     0  raflux 0/1     no/yes - write angular fluxes to file raflux(unimplemented)
*                     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  isdenx 0/1/n - none/x density vector/full matrix
*                                    0  isdeny 0/1 no/yes - use y density vector
*                                    0  igan   source anisotropy
*                                    0  isorse number of source vectors input
*                                    0  isorsx number of sourcx vectors input
*                                    0  isorsy number of sourcy vectors input
*                                    0  isorsf number of sourcf vectors input
*
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
```

```
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*
*                                    ...parameters from block i...
*
*                          9  igeom   9  triangular x-y
*                          2  ngroup  number of energy groups
*                          4  isn     angular quadrature order
*                          3  mt      number of permanent materials
*                          2  nzone   number of zones
*                         12  it      number of triangles per row
*                          6  jt      number of rows of triangles
*
*
*
*
*********************************************************************************************************************************
*********************************************************************************************************************************


*********************************************************************************************************************************
*********************************************************************************************************************************


        29 words lcm required 2,1.1
*
*
*                                    ...material assignments to zones...
*
*
*******************************
*key start matls to zones     *
*******************************
*
*                  zone cross section = sum over matls in the zone of (matl cross section)*( c0 + c1*cmod )
*                                             with
*                                          cmod =  0.000000E+00
*
*                              entry   zone      material        c0              c1
*                                    no. name    no. name
*
*                                1     1 CORE      2 FUEL      3.500000E-01   0.000000E+00
*                                2     1 CORE      3 SODIUM    4.000000E-01   0.000000E+00
*                                3     1 CORE      1 STEEL     2.500000E-01   0.000000E+00
*                                4     2 REFLEC    3 SODIUM    7.000000E-01   0.000000E+00
*                                5     2 REFLEC    1 STEEL     3.000000E-01   0.000000E+00
*
*
*


    scm storage summary...

    total scm required for this problem   1623
    maximum scm available                40000


    lcm storage summary...

    total lcm required for this problem        1418
    maximum lcm specified                    140000


        1418 words lcm required 2,1.2


zero flux




***** s 4 this is the constant set iquad=1


                  mu            eta            xi             phi            weight

      1     0.81443825E+00  0.47021614E+00  0.33998104E+00     30.0       0.10869086E+00
      2     0.44026491E+00  0.25418706E+00  0.86113631E+00     30.0       0.57975809E-01
      3     0.00000000E+00  0.94043229E+00  0.33998104E+00     90.0       0.10869086E+00
      4     0.00000000E+00  0.50837413E+00  0.86113631E+00     90.0       0.57975809E-01
      5    -0.81443825E+00  0.47021614E+00  0.33998104E+00    150.0       0.10869086E+00
      6    -0.44026491E+00  0.25418706E+00  0.86113631E+00    150.0       0.57975809E-01
*********************************************************************************************************************************
*********************************************************************************************************************************
*
*                      ...cross section related data from file maccxs  03/24/17:08: version  1 ...
*
*********************************************************************************************************************************
*
*
*  1 STEEL     2 FUEL      3 SODIUM
*
*
*
*********************************************************************************************************************************
```

```
    left number indicates y-coordinate.

  60.00                   1-----2-----3-----4-----5-----6-----7
row
 6
  50.00                1-----2-----3-----4-----5-----6-----7
row                          2
 5                     2   2
  40.00             1-----2-----3-----4-----5-----6-----7
row                    2   2   2
 4                  1   2   2   2
  30.00          1-----2-----3-----4-----5-----6-----7
row                 1   1   2   2
 3               1   1   1   2
  20.00       1-----2-----3-----4-----5-----6-----7
row              1   1   1   2
 2            1   1   1   2   2
  10.00    1-----2-----3-----4-----5-----6-----7
row           1   1   1   2   2
 1         1   1   1   1   2
   0.00 1-----2-----3-----4-----5-----6-----7
*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=
*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=
*
*
* epsi   1.00E-04
* norm   1.00E+00
* bhgt   9.00E+01
*
*
*
* time  outer          k-eff                  max ptwise  dominance ratio
* (sec) no. inners      ev         lambda-1  fiss change estimate    used
   3.2   0    0     1.00000000 -1.000E+00 1.000E+00  0.000E+00  0.000E+00 ###
   3.2   1   16     0.81515341 -1.848E-01 8.996E-01  0.000E+00  0.000E+00 ###
   3.2   2   17     1.08858116  3.354E-01 3.508E-01  0.000E+00  0.000E+00 ###
   3.3   3   17     1.11496484  2.424E-02 9.750E-02  2.167E-01  0.000E+00 ###
   3.3   4   16     1.12373763  7.868E-03 4.889E-02  4.974E-01  0.000E+00 ###
   3.4   5   14     1.12948036  5.110E-03 2.604E-02  4.974E-01  4.974E-01 ###
   3.4   6   13     1.13184986  2.098E-03 9.496E-03  5.189E-01  4.974E-01 ###
   3.4   7   14     1.13287543  9.061E-04 2.697E-03  5.420E-01  4.974E-01 ###

                              **** begin outer iteration   8 ****
                              inner            max ptwise
                  group    iterations          flux error
                    1          5               3.367E-05
                    2         10               5.941E-06
   3.5     8   15   1.13317541  2.648E-04  8.526E-04  5.420E-01  5.420E-01 ###

                              **** begin outer iteration   9 ****
                              inner            max ptwise
                  group    iterations          flux error
                    1          5               9.431E-06
                    2         12               1.067E-06
   3.5     9   17   1.13335922  1.622E-04  3.628E-04  5.824E-01  5.420E-01 ###

                              **** begin outer iteration  10 ****
                              inner            max ptwise
                  group    iterations          flux error
                    1          5               4.716E-06
                    2         13               1.294E-07


                                        ...interface file rtflux written..

   3.6    10   18   1.13331547 -3.860E-05  7.279E-05  5.419E-01  5.420E-01 ###
*
*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=
*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=
*
*
*                                        ...interface file snccns written..


twohex time mins   5.9543E-02
**************************************************************************************************************************************
```

```
*****************************************************************************************************************************
*
*                                edit run on 03/24/95 with solver version 01-25-95*product release 2.6mx    machine e
*
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                                ...edit output...
*
*****************************************************************************************************************************
*
*
*                                             ...block vi - edit specification data...
*
*
************************
*key start edit output *
************************


*****************************************************************************************************************
*
*
* cross section balancing (balxs.ne.0)
*       or
* transport correction (trcor=diag, cesaro, or bhs)
* will NOT be reflected in edits
*
*
*****************************************************************************************************************


*                                          ...input control integers...
*
*                  1  pted      0/1     no/yes - point edits desired
*                  0  zned      0/1     no/yes - zone edits desired
*
*                  0  ajed      0/1     direct/adjoint edit (use rtflux/atflux file)
*
*                  0  igrped    0/1/2/3 print totals only/print broad groups only/same as 1/print all groups and totals
*                  0  byvolp    0/1     no/yes - multiply point reaction rates by mesh volumes
*                  0  rzflux    0/1     no/yes - write the rzflux file (zone average flux file)
*                  0  rzmflx    0/1     no/yes - write the rzmflx file (zone average flux moments file)
*
*
*                                          ...floating parameters...
*
*          0.000000E+00  power     0/p     no/normalize all results, including flux files, to p megawatts
*          2.100000E+02  mevper    mev per fission (default: 210 mev)
*
*
*
*                                       ...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 broad groups
*
*
*
*                                       ...space related edit information...
*
*                                         72  number of points to edit
*                                          0  number of zones to edit
*                                          0  0/1  no/yes  density factors were input
*                                          1  0/1/2/3  no/solver/edit/both  zones mass edit
*                                                (requires atomic weights to be present)
*
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*                                                ...edit output...
*
*****************************************************************************************************************************
*
*
*                                       ...edit specification data (continued)...
*
*
*
*                                       ...description of cross section edits...
*
*                                isotope  no.      material  no.      constituent  no.
*                                -------  ---      --------  ---      -----------  ---
*
*                                -none-            resdnt             -none-
*
*
*                                     reaction rates will be formed for each
*                                                of the above
*                                   using the cross section types shown below
*
*                                                        position
*                                                type      no.
*                                               ------     ---
*                                               chi         1
*                                               nusigf      2
*                                               total       3
*                                               abs         4
*                                               edit1       5
*
*****************************************************************************************************************************
*****************************************************************************************************************************
*****************************************************************************************************************************
```

```
*****************************
*key start     materials     *
*****************************
*                                  ****************************
*                                  *                          *
*                                  *       reaction rates     *
*                                  *           from           *
*                                  *         materials        *
*                                  *                          *
*                                  ****************************
*
*
*                                ...point edit for the sum of the neutron groups ...
*
*
*************************************************************************************************************************************
*************************************************************************************************************************************
*
*    point chi      edit for k-plane  1, and broad group  3, of macroscopic cross sections.
*
* all entries of this array equal  0.00000E+00
*************************************************************************************************************************************
*************************************************************************************************************************************
*
*    point nusigf   edit for k-plane  1, and broad group  3, of macroscopic cross sections.
*
*
* component number  1
*
*        x-mesh    1   x-mesh    2   x-mesh    3   x-mesh    4   x-mesh    5   x-mesh    6   x-mesh    7   x-mesh    8
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 6.496816E-04 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 1.129600E-03 8.782451E-04 8.123235E-04 5.532799E-04 4.604951E-04 0.000000E+00 0.000000E+00 0.000000E+00
*     2 1.500788E-03 1.310747E-03 1.221545E-03 9.732105E-04 8.123235E-04 5.532799E-04 0.000000E+00 0.000000E+00
*     1 1.704111E-03 1.601047E-03 1.500788E-03 1.310747E-03 1.129600E-03 8.782451E-04 6.496816E-04 0.000000E+00
*
*        x-mesh    9   x-mesh   10   x-mesh   11   x-mesh   12
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     2 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     1 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*************************************************************************************************************************************
*************************************************************************************************************************************
*
*    point total    edit for k-plane  1, and broad group  3, of macroscopic cross sections.
*
*
* component number  1
*
*        x-mesh    1   x-mesh    2   x-mesh    3   x-mesh    4   x-mesh    5   x-mesh    6   x-mesh    7   x-mesh    8
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 3.096076E-03 1.664602E-03 1.554044E-03 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 1.070705E-02 4.765247E-03 4.427475E-03 2.648309E-03 2.205208E-03 1.062666E-03 7.925773E-04 0.000000E+00
*     3 1.835132E-02 1.431912E-02 1.325457E-02 9.131630E-03 7.636400E-03 3.046368E-03 2.205208E-03 1.062666E-03
*     2 2.434128E-02 2.126977E-02 1.982762E-02 1.582306E-02 1.325457E-02 9.131630E-03 4.427475E-03 2.648309E-03
*     1 2.762950E-02 2.596175E-02 2.434128E-02 2.126977E-02 1.835132E-02 1.431912E-02 1.070705E-02 4.765247E-03
*
*        x-mesh    9   x-mesh   10   x-mesh   11   x-mesh   12
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     2 1.554044E-03 0.000000E+00 0.000000E+00 0.000000E+00
*     1 3.096076E-03 1.664602E-03 0.000000E+00 0.000000E+00
*************************************************************************************************************************************
*************************************************************************************************************************************
*
*    point abs      edit for k-plane  1, and broad group  3, of macroscopic cross sections.
*
*
* component number  1
*
*        x-mesh    1   x-mesh    2   x-mesh    3   x-mesh    4   x-mesh    5   x-mesh    6   x-mesh    7   x-mesh    8
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 7.786503E-06 4.204682E-06 3.921565E-06 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 3.972334E-04 1.207441E-05 1.121578E-05 6.648248E-06 5.529812E-06 2.673369E-06 2.003130E-06 0.000000E+00
*     3 6.844552E-04 5.333548E-04 4.935612E-04 3.386031E-04 2.826669E-04 7.660331E-06 5.529812E-06 2.673369E-06
*     2 9.084230E-04 7.936439E-04 7.397584E-04 5.899858E-04 4.935612E-04 3.386031E-04 1.121578E-05 6.648248E-06
*     1 1.031272E-03 9.689775E-04 9.084230E-04 7.936439E-04 6.844552E-04 5.333548E-04 3.972334E-04 1.207441E-05
*
*        x-mesh    9   x-mesh   10   x-mesh   11   x-mesh   12
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     3 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     2 3.921565E-06 0.000000E+00 0.000000E+00 0.000000E+00
*     1 7.786503E-06 4.204682E-06 0.000000E+00 0.000000E+00
*************************************************************************************************************************************
*************************************************************************************************************************************
*
*    point edit1    edit for k-plane  1, and broad group  3, of macroscopic cross sections.
*
*
* component number  1
*
*        x-mesh    1   x-mesh    2   x-mesh    3   x-mesh    4   x-mesh    5   x-mesh    6   x-mesh    7   x-mesh    8
*
*     6 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     5 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
*     4 2.121738E-04 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
```

```
*     3 3.684618E-04  2.865588E-04  2.650667E-04  1.807132E-04  1.504682E-04  0.000000E+00  0.000000E+00  0.000000E+00
*     2 4.894711E-04  4.275086E-04  3.984238E-04  3.174702E-04  2.650667E-04  1.807132E-04  0.000000E+00  0.000000E+00
*     1 5.557677E-04  5.221605E-04  4.894711E-04  4.275086E-04  3.684618E-04  2.865588E-04  2.121738E-04  0.000000E+00
*
*         x-mesh    9   x-mesh   10   x-mesh   11   x-mesh   12
*
*     6 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
*     5 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
*     4 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
*     3 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
*     2 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
*     1 0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*
*                                                    run highlights
*****************************
*key start run highlights    *
*****************************
*                                        ****************************************
*                                        *                                      *
*                                        *   all modules are tentatively go.    *
*                                        *    interface file geodst written.    *
*                                        *      cross sections from cards.      *
*                                        *   interface mixing files written.    *
*                                        *    interface file asgmat written     *
*                                        *   xs files macxs,srcedt written.     *
*                                        *    interface file solinp written.    *
*                                        *    interface file editit written.    *
*                                        *       start edit execution.          *
*                                        *          edits completed.            *
*                                        *                                      *
*                                        ****************************************
*
*
*                                                  storage and timing history
*                                  ************************************************************
*                                  *          scm     scm     lcm     lcm     cpu     sysio   *
*                                  * module  words   limit   words   limit  seconds  seconds  *
*                                  * ------  -----   -----   -----   -----  -------  -------  *
*                                  *    0        0       0       0       0     3.8      0.4    *
*                                  *  100     3662   40000       0       0     3.1      0.2    *
*                                  *  101        0       0       0       0     0.6      0.0    *
*                                  *  102     1092   40000      79  140000     0.0      0.0    *
*                                  *  103       50   40000       0       0     0.0      0.0    *
*                                  *  104     2066   40000       0       0     0.1      0.0    *
*                                  *  105        0       0       0       0     0.0      0.0    *
*                                  *  106        0       0       0       0     0.0      0.0    *
*                                  *  107      191   40000      83  140000     0.1      0.0    *
*                                  *  108      547   40000       7  140000     0.0      0.0    *
*                                  *  109     3677   40000       0       0     0.1      0.0    *
*                                  *  112        0       0       0       0     0.0      0.0    *
*                                  *  200     1623   40000    1418  140000     0.5      0.1    *
*                                  *   21        0       0       0       0     0.0      0.0    *
*                                  *   22        0       0       0       0     0.0      0.0    *
*                                  *   23      259   40000       0       0     0.0      0.0    *
*                                  *   24        0       0       0       0     0.0      0.0    *
*                                  *   25        0       0       0       0     0.0      0.0    *
*                                  *   26        0       0       0       0     0.4      0.0    *
*                                  *   27        0       0       0       0     0.0      0.0    *
*                                  *   28        0       0       0       0     0.0      0.0    *
*                                  *   29      661   40000       0       0     0.0      0.0    *
*                                  *  300        0       0       0       0     0.3      0.1    *
*                                  *  301     1660   40000     721  140000     0.2      0.1    *
*                                  *  302        0       0       0       0     0.0      0.0    *
*                                  *  400        0       0       0       0     0.0      0.0    *
*                                  ************************************************************
*
*
*                                                ...execution terminated...
*
*
*****************************************************************************************************************************
*****************************************************************************************************************************
```

APPENDIX B: OPERATING SYSTEM SPECIFICS

UNIX/UNICOS Execution

On UNIX or UNICOS systems, the input is on STDIN and the printed output is on STDOUT. Thus, the user will normally cause execution of the program with the command:

dant.x < *odninp* > *odnout*

where dant.x is the name of the executable file, *odninp* is the user's choice for a name for the input file, and *odnout* is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is dant.x.

STDERR contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDERR are any error messages.

Library Search Path

Most files read or written by TWOHEX are in the current UNIX working directory. Some forms of cross-section files may be kept in other directories. By setting the environment variable SNXSPATH, the user may specify an ordered set of alternate directories in which the program should look for the named files. As an example, if an ISOTXS file is in the directory, `/usr/tmp/xs`, then the following command can be used

```
setenv SNXSPATH /usr/tmp/xs
```

and TWOHEX will then look in that named directory for the library. The search path for each of the possible libraries is given in Table 5.2.

Table 5.2 UNIX Search Path

LIB	SEARCH PATH
MACRXS	Current Working Directory (CWD).
GRUPXS	**SNXSPATH**, then CWD.
ISOTXS	**SNXSPATH**, then CWD.
BXSLIB	**SNXSPATH**, then CWD, but see text below.
ODNINP	None, the library is contained in the input file.
MACBCD	CWD
XSLIBB	CWD
MENDF[a]	Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.
MENDFG[b]	
XSLIB	**SNXSPATH**, then CWD
other	For any name other than those above, the program will assume the form is XSLIB and search for it in **SNXSPATH**, then CWD.

a. Available only at Los Alamos.

b. Available only at Los Alamos.

SNXSPATH can be used to protect an input BXSLIB file from being overwritten. See the discussion on page 5-37.

THREEDANT USER'S GUIDE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

6

XTM — Transport Methods Group

Los Alamos
National Laboratory

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

An Affirmative Action/Equal Opportunity Employer

This work was supported by the US Department of Energy.

DISCLAIMER

USER'S GUIDE FOR THREEDANT: A CODE PACKAGE FOR THREE-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT

by
Ray E. Alcouffe, Randal S. Baker, Forrest W. Brinkley, and Duane R. Marr

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The THREEDANT code package is a modular computer program designed to solve the three-dimensional, time-independent, multigroup discrete-ordinates form of the Boltzmann transport equation.[1,2] It is a three-dimensional version of the one-dimensional code, ONEDANT.[3]

THREEDANT™ is based on the modular construction of the DANTSYS™ code system package. This modular construction separates the input processing, the transport equation solving, and the postprocessing, 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 used by all the codes in the package. Different solution techniques are invoked simply by executing different SOLVER modules in the package. This SOLVER choice is automatically made by the INPUT module through an analysis of the input stream.

The THREEDANT code is simply the DANTSYS code system package with a three-dimensional SOLVER module.

Some of the major features included in the THREEDANT package are:

1. a free-field format text input capability, designed with the user in mind,

2. standardized data and file management techniques as defined[4] and developed by the Committee on Computer Code Coordination (CCCC); both sequential file and random-access file handling techniques are used,

3. the use of a diffusion synthetic acceleration scheme[5] to accelerate the iterative process in the SOLVER module,

4. direct (forward) or adjoint calculational capability,

5. x-y-z and r-z-θ geometry options,

6. arbitrary anisotropic scattering order,

7. vacuum, reflective, periodic, white, or surface source boundary condition options,

8. inhomogeneous (fixed) source or k_{eff} calculation options, as well as time-absorption (α), nuclide concentration, or dimensional search options,

---

DANTSYS and THREEDANT are trademarks of the Regents of the University of California, Los Alamos National Laboratory.

9. "diamond-differencing" and adaptive weighted diamond differencing (AWDD) for solution of the transport equation,

10. a diffusion solver that uses the multigrid method,[6]

11. user flexibility in using either ASCII text or sequential file input,

12. user flexibility in controlling the execution of both modules and submodules, and

13. extensive, user-oriented error diagnostics.

DOCUMENTATION FOR THREEDANT USAGE

The documentation described here constitutes a complete manual for the use of the THREEDANT code.

Included are two general categories of information. The first category is in this User's Guide and is oriented towards preparing input to the code. The second category is of a background, reference, conceptual, or theoretical nature and is intended primarily for the novice or first time user; an experienced user generally needs only this User's Guide.

What Is In This User's Guide

This User's Guide is a chapter from the much larger DANTSYS system document. This Guide provides the ASCII text input specifications for THREEDANT.

The guide is intended to serve as a complete input manual for two classes of user. Special, succinct sections containing summaries and compact tables are intended for the advanced user in order to make his input preparation more efficient. The main body of the guide concerns itself with descriptions of the input and should be sufficient for the user familiar with discrete ordinates concepts. Novice users may find other chapters of the document necessary.

This Guide first gives an overview of the input block order required by the code.

Next is a "mini-manual" in which are listed all the names of available input arrays arranged by input block. Definitions of input arrays are not given, as the names are suggestive, but expected types and sizes are provided. This mini-manual is very useful to the user as a quick check for completeness, a quick reference to type and size, and as an index into the more detailed array descriptions that follow. For the experienced user, the mini-manual is frequently all that is needed to prepare a complete input deck.

Following the mini-manual are reference sections describing in detail all the input parameters and arrays.

Appendix A provides a sample THREEDANT problem with explanation of the output for the user.

Lastly, Appendix B details operating system specifics, including how to effect an execution of the code.

Information of a reference, background, or theoretical nature that the first time user may need may not be found in this User's Guide, but the user will encounter liberal references to other chapters of this document for that sort of information.

What Is Available Elsewhere

In addition to this User's Guide, the user, especially the first time user, may find the information below described in other chapters of this document pertinent. For even greater detail on some of the general items, particularly the methods items, the user should look at Ref. 7.

The chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1 discusses in more detail the geometry and solver concepts and their related input. If the User's Guide proves insufficient for your needs, look in this chapter. Among the many sections of the chapter are ones on the input of inhomogeneous sources and a discussion of eigenvalue searches. There is also more detail on the Block-I input.

A discussion of how the EDIT module works and more detail on preparing the input is given in the chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 serves as the reference manual for the free-field input (rules, format, and operators) used in this code. That chapter is summarized in this guide, but should the summary prove inadequate, the user is referred there for full details.

The chapter "CROSS-SECTION LIBRARIES" starting on page 10-1 gives details of the many library formats available to THREEDANT, including sections on how to prepare your own card-image (or text) libraries.

The chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1 describes the mixing concepts in detail and shows some examples.

Next is the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. That chapter describes the theoretical basis for the THREEDANT code as well as the other codes in the DANTSYS system package.

In the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure" starting on page 13-1 is shown a brief 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. In particular, the code package structure must be understood in order to make up input for piecewise executions of the code that are possible with controls that are part of the input in Block-I.

Error diagnostics that the user might encounter are found in the chapter "ERROR MESSAGES" starting on page 14-1. Several examples of input errors and the resulting error messages are provided for the user.

The chapter "FILE DESCRIPTIONS" starting on page 15-1 is a reference that describes all the files used by the package. Included is a detailed description of the file structure of the code dependent, binary, sequential interface files generated by and used in the

DANTSYS code package. Also included are descriptions of any other files produced or used by the package, both binary and text. In some cases, this may simply be a reference to a more comprehensive document, such as the file descriptions for the CCCC standard interface files.

THREEDANT INPUT OVERVIEW

Input Block Order

The full THREEDANT input consists of a title line section, followed by six blocks of free field input. The title line section is not free field. Any input referred to as a block uses the free field input form.

Block-I consists of basic control and dimensional information that allows efficient packing of the array data. This information also allows checking of the lengths of arrays supplied by interface files.

Block-II contains the geometric information.

Block-III consists of the nuclear data specifications.

Block-IV contains mixing information.

Block-V contains the rest of the input needed for specifying the flux calculation.

And lastly, Block-VI contains the edit (i.e., report writing) specifications.

If a text cross-section library is to be included in the input deck, it should be placed between Blocks III and IV. THREEDANT supports many library formats and so the library may or may not be in free field format depending upon the option chosen.

A full input would then look like that diagrammed on the following page.

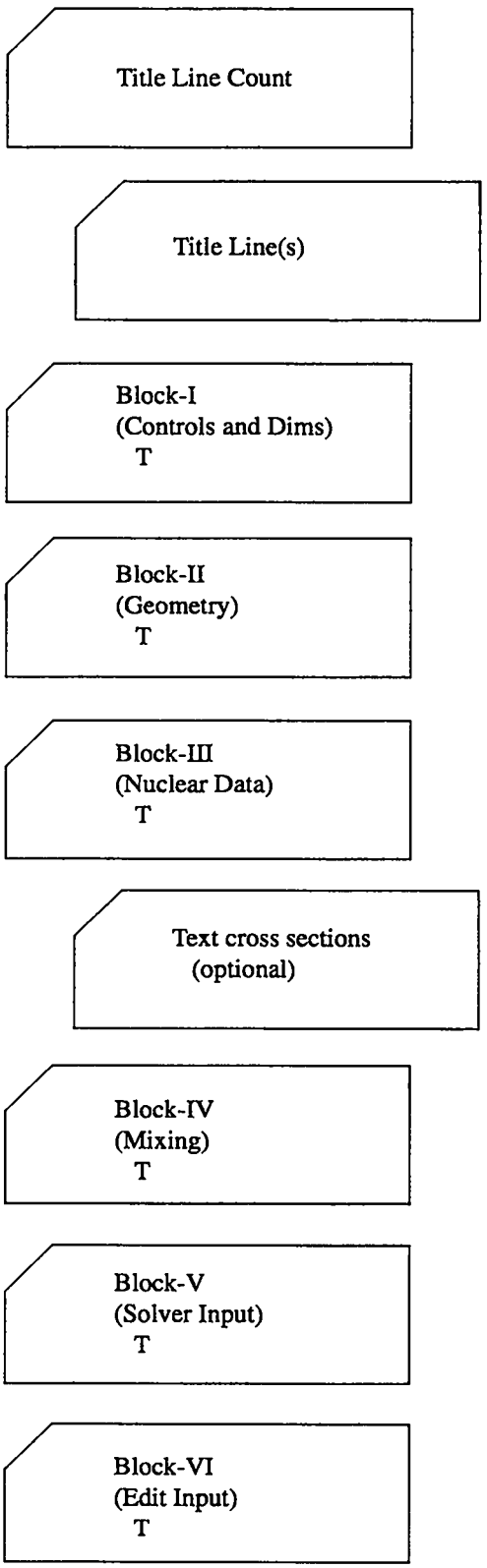


Figure 6.1 THREEDANT Input Order

Free Field Input Summary

The chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 is summarized here for quick reference.

There are four basic input quantities in the free field input used in THREEDANT; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is briefly described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array. "To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, might look like:

```
SOURCX= 0 0 0 1.1 1.1 0 0 0 0 0
```

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Single value input variables are treated as arrays of unit length.

Numeric Data Items

Numeric data items follow a Fortran input convention. For example, all of the following are valid entries for the number ten:

```
10, 1.0+1, 1E1, 10.0
```

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a Fortran variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters and blanks may be included if the data item is surrounded by double quotes. Operators may NOT be used with character data items.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this, the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described later, in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input.

The data operators are specified in the general form

n *O* d

where:

n is the "data numerator," either an integer or a blank;
O is any one of the "data operator" characters shown below; and
d is a "data entry" (may be blank for some operators).

Note: The "data operator" character must be appended to the "data numerator."

Using operators, the SOURCX input described above could more succinctly be given as:

SOURCX= 0 0 0 2R 1.1 F0

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO[8] operators may be used in numeric entry.

A table of the most used operators is given next including brief descriptions. For full descriptions of these and a complete list of all the available operators, including the more esoteric ones, the user is referred to "FREE FIELD INPUT DETAILS" on page 9-13.

Frequently Used Operators

Operator[a]	Functionality
nR d	REPEAT the data item d, n times.
nI d	INTERPOLATE (linear) n data items between data item d and the next data item.
nC d	SCALE (multiply) the n previous entries by d.
F d	FILL the rest of the data string with the data item d.
nY m	STRING REPEAT. Repeat the previous m strings, n times.
nL d	INTERPOLATE LOGARITHMICALLY n data items between d and the next d.
nZ	ZERO. Enter the value zero n successive times.
nS	SKIP. Skip the next n data items.
nQ m	SEQUENCE REPEAT. Enter the last m entries, n more times.
nG m	SEQUENCE REPEAT WITH SIGN CHANGE. Same as the Q option but the sign of the m entries is changed every repeat.
nN m	SEQUENCE REPEAT INVERT. Same as the Q option but the order of the m entries is inverted each repeat.
nM m	SEQUENCE REPEAT INVERT WITH SIGN CHANGE. Same as N option but the sign is also changed every repeat.
nX	COUNT CHECK. Causes code to check the number of entries in the current string so far, against the number n.

a. The operator character must always be appended directly to n. d or m need not be immediately adjacent to the operator character.

MINI-MANUAL Introduction

On the following few pages is given a complete list of the input names, expected array sizes, and order within the array. No description of the array contents is given in this MINI-MANUAL as full details are given in later sections. The MINI-MANUAL is intended to serve as a quick reference for the knowledgeable user.

In both the MINI-MANUAL and in the detailed sections which follow, a shorthand form is used to indicate the size and order of the array that the code expects. This information is enclosed in square brackets immediately after the array name. Essential features are:

1. A single entry in the brackets is the array length.

2. No brackets at all indicates a simple variable (i.e., an array of unit length).

3. A dash (-) in the brackets indicates an arbitrary length.

4. A semicolon (;) indicates that the input for that array is expected in strings. To the left of the semicolon is the string length. To the right of the semicolon is the number of strings in the array.

5. If the number of strings is shown as a product, the order is important. The left-most quantity must be exhausted first, then, the next one to the right is varied. For example, the array name for the full spatial source distribution is shown as:

   SOURCF [IT;JT*KT*NMQ]

   where - IT is the number of fine meshes in the X-direction, JT is the number of fine meshes in the Y-direction, KT is the number of fine meshes in the Z-direction, and NMQ is the number of input source moments. For this array, the first string is composed of the P_0 source values for each x mesh point in the first y mesh in the first z layer. The next string is the P_0 source values in the second y mesh in the first z layer. This process is repeated for all JT y meshes of the first z layer. Then repeat for each of the remaining z layers. Then starting again with the first y mesh in the first z layer, the P_1 source values for each x mesh are given. After all P_1 values are given, the P_2 values follow. Continue until all NMQ moments are specified.

Note: Usually, values for the quantities within brackets will have already been specified in the input. Sometimes, however, a quantity is derived from the array input itself. For instance, in this particular case, NMQ is not an input quantity; rather, the code counts the number of strings and then, knowing JT, KT, and NGROUP, deduces what NMQ must have been.

MINI MANUAL

Title Line Control
(3I6 Format)
NHEAD,NOTTY,NOLIST

Title Line(s)

(IF NHEAD>0)

Block-I:Controls & Dimensions

IGEOM
NGROUP
ISN
NISO
MT
NZONE
IM
IT
JM
JT
KM
KT

MAXLCM
MAXSCM

-- -- -- -- --

NOSOLV
NOEDIT

NOGEOD
NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM

T

Block-II:Geometry

XMESH [IM+1]
YMESH [JM+1]
ZMESH [KM+1]

XINTS [IM]
YINTS [JM]
ZINTS [KM]

ZONES [IM;JM*KM]

T

Block-III: Cross Sections

LIB
  valid: *ODNINP*
  XSLIB
  ISOTXS
  GRUPXS
  BXSLIB
  MACRXS
  MACBCD
  XSLIBB
  (local)*MENDF*
  (local)*MENDFG*
  alternate XSLIB name

WRITMXS
  valid: *MACBCD*
  XSLIBB
  XSLIBF
  XSLIBE

LNG
BALXS
NTICHI
CHIVEC [NGROUP]
LIBNAME
-- -- -- -- -- -- --
Rest of this block is needed only for text libraries.

MAXORD
IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD (default:1)
NAMES [NISO]
EDNAME [IHT-3]
NTPI [NISO]
VEL [NGROUP]
EBOUND [NGROUP+1]

T

iff LIB= ODNINP, insert ASCII text cross sections here

Block-IV: Mixing

MATLS [-;MT]
ASSIGN [-;NZONE]

PREMIX [-;-]

ASGMOD [-;-]
CMOD

FMMIX

MATNAM [MT]
ZONNAM [NZONE]
MATSPEC [-]
  valid: *ATFRAC*
  WTFRAC
  ATDEN

ATWT [-]

T

Block-V: Solver

IEVT
ISCT
ITH
IBL
IBR
IBT
IBB
IBFRNT
IBBACK

EPSI
IITL
IITM
OITM
ITLIM
NOSIGF
KCALC

Solver (continued)

--- Output Controls ---

FLUXP
XSECTP
FISSRP
SOURCP
ANGP
BALP
RAFLUX
RMFLUX

--- Miscellaneous ---

TRCOR
valid: *DIAG*
BHS
CESARO
NO
NORM

CHI [NGROUP;M]

DEN [IT;JT*KT]
--- or ---
DENX [IT], DENY[JT], DENZ[KT]

--- Spatial Discretization ---

WDAMP [NGROUP]

--- Quadrature -------

GRPSN [NGROUP]
IQUAD
WGT [MM]
MU [MM]
ETA [MM]

Solver (continued)

--- Flux Guess -------

INFLUX

--- Searches ---------

IPVT
PV
EV
EVM
XLAL
XLAH
XLAX
POD

XM [IM]
YM [JM]
ZM [KM]

Solver (continued)

----Volumetric Source----

INSORS

SOURCE [NGROUP;NMQ]
--- or ---
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ] and
SOURCZ [KT;NMQ]
--- or ---
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ] and
SOURCZ [KT;NMQ] and
SOURCE [NGROUP;NMQ]
--- or ---
SOURCF [IT;JT*KT*NGROUP*NMQ]
--- or ---
SOURCF [IT;JT*KT*NMQ] and
SOURCE [NGROUP;NMQ]

-----Boundary Source-----

Options 1: ----
SILEFT [NGROUP;JT*KT]
SIRITE [NGROUP;JT*KT]
SIBOTT [NGROUP;IT*KT]
SITOP [NGROUP;IT*KT]
SIFRNT [NGROUP;IT*JT]
SIBACK [NGROUP;IT*JT]

Options 2: ----
SALEFT [MM*4;NGROUP*JT*KT]
SARITE [MM*4;NGROUP*JT*KT]
SABOTT [MM*4;NGROUP*IT*KT]
SATOP [MM*4;NGROUP*IT*KT]
SAFRNT [MM*4;NGROUP*IT*JT]
SABACK [MM*4;NGROUP*IT*JT]

Options 3a: ----
BSLFTG [NGROUP]
BSLFTY [JT]
BSLFTZ [KT]
BSLFTA [MM*4]

BSRITG [NGROUP]
BSRITY [JT]
BSRITZ [KT]
BSRITA [MM*4]

Solver (continued)

----Boundary Source (cont'd)----

BSBOTG [NGROUP]
BSBOTX [IT]
BSBOTZ [KT]
BSBOTA [MM*4]

BSTOPG [NGROUP]
BSTOPX [IT]
BSTOPZ [KT]
BSTOPA [MM*4]

BSFRNG [NGROUP]
BSFRNX [IT]
BSFRNY [JT]
BSFRNA [MM*4]

BSBAKG [NGROUP]
BSBAKX [IT]
BSBAKY [JT]
BSBAKA [MM*4]

Options 3b: ----
BSLFTG [NGROUP]
BSLFTYZ [JT;KT]
BSLFTA [MM*4]

BSRITG [NGROUP]
BSRITYZ [JT;KT]
BSRITA [MM*4]

BSBOTG [NGROUP]
BSBOTXZ [IT;KT]
BSBOTA [MM*4]

BSTOPG [NGROUP]
BSTOPXZ [IT;KT]
BSTOPA [MM*4]

BSFRNG [NGROUP]
BSFRNXY [IT;JT]
BSFRNA [MM*4]

BSBAKG [NGROUP]
BSBAKXY [IT;JT]
BSBAKA [MM*4]

SOLVER (continued)

----Boundary Source (cont'd)----

Options 3c: ----
BSLFTG [NGROUP]
BSLFTYZ [JT;KT*MM*4]

BSRITG [NGROUP]
BSRITYZ [JT;KT*MM*4]

BSBOTG [NGROUP]
BSBOTXZ [IT;KT*MM*4]

BSTOPG [NGROUP]
BSTOPXZ [IT;KT*MM*4]

BSFRNG [NGROUP]
BSFRNXY [IT;JT*MM*4]

BSBAKG [NGROUP]
BSBAKXY [IT;JT*MM*4]

Options 3d: ----
BSLFTG [NGROUP]
SALEFT [MM*4;JT*KT]

BSRITG [NGROUP]
SARITT [MM*4;JT*KT]

BSBOTG [NGROUP]
SABOTT [MM*4;IT*KT]

BSTOPG [NGROUP]
SATOPT [MM*4;IT*KT]

BSFRNG [NGROUP]
SAFRNT [MM*4;IT*JT]

BSBAKG [NGROUP]
SABAKT [MM*4;IT*JT]

Block-VI: EDIT

PTED
ZNED

POINTS [K], K≤IT*JT*KT
EDZONE [IT;JT*KT]

EDXS [K], K≤NEDT
RESDNT
EDISOS [K], K≤NISO
EDCONS [K], K≤NISO
EDMATS [K], K≤MT
XDF [IT]
YDF [JT]
ZDF [KT]

RSFE [NGROUP;-]
RSFX [IT;-]
RSFY [JT;-]
RSFZ [KT;-]
RSFNAM [-]

ICOLL [K], K≤NGROUP
IGRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

PRPLTED
IPLANE
JPLANE
KPLANE

T

THREEDANT INPUT DETAILS

Introduction

The following pages of this section give details for each of the input arrays. All valid THREEDANT arrays are discussed in this section in detail complete enough to form the input.

However, the beginning user, particularly one unfamiliar with discrete-ordinates codes, may find that he is missing some information of a background nature. See "What Is Available Elsewhere" on page 6-14 for that.

First, here are a few general instructions:

1. All six of the input blocks are normally included. Block-I is always required but any of the other five blocks may be omitted under the proper conditions. The input module reads each block in turn and from it generates one or more binary interface files. The interface files drive the SOLVER and EDIT modules. Thus, if the user wants no edits, the Block-VI input may be omitted. Then with no interface file, the EDIT module will not be executed. Alternatively, if the interface file is available from another source, the corresponding block of input may be omitted. For instance, Block-II describes the geometry. The input module normally writes this information to the GEODST interface file. If the GEODST file is available from another source or a previous run, the Block-II input may be omitted.

2. A general theme of the THREEDANT input is that arrays that are not needed are not entered. Presence of an array indicates that it should be used. Thus, for example, if the density array is entered (DEN array), the cross section at each mesh interval will be modified accordingly. No separate switch need be set to say that the calculation should be done. To eliminate the density modification, simply remove the DEN array from the input or comment it out.

3. The arrays, in general, are grouped in the input instructions according to function. Thus, for example, the input arrays for the volumetric source are found in a single table, or grouping, of input.

4. Groupings of input data may be marked as "Required" or "Optional" in order to guide the user and speed navigation through the input instructions.

   "Required" means that at least one of the arrays in the grouping must be entered. Thus, you must read through the grouping and enter at least one of the arrays found there.

Groupings marked "Optional" may be skipped if the subject is inappropriate. Thus, using the previous example, if one has no volumetric source, one simply skips to the next grouping of input; there is no need to read about any of the arrays within the volumetric source grouping.

Arrays in groupings not marked as "Required" or "Optional" should be reviewed. These groupings contain arrays of vital data that are used in every calculation, but have default values. Thus, although you may not make any input to these arrays and they are in that sense optional, you must concern yourself with them to ensure that the default values are what is intended.

5. Input arrays may also be marked individually. If not marked, they inherit the marking of the grouping in which they are contained. Thus, an unmarked array in a "Required" grouping is required input and you must enter that array. An unmarked array in an "Optional" grouping is optional.

   You may encounter a "Required" array within an "Optional" grouping. That means that if you decide to invoke the option represented by that grouping, you must input that particular array. For example, if you want user defined response function reaction rates calculated, you must input the RSFE array.

   All arrays within unmarked groupings are optional. However, values in these arrays may be used by the code, so you should concern yourself with the default values if you choose not to enter a value.

6. Unless specifically noted otherwise, the default on all numeric inputs is zero.

7. In an adjoint run, none of the groupwise input arrays should be inverted. The code will externally identify all groups by the physical group number, not by the calculational group number (the calculational group number is in inverse order). Thus, the user interface should be consistently in the physical group order.

8. The use of information within square brackets to indicate the size of arrays and strings and the order within those arrays is the same as described in "MINI-MANUAL Introduction" on page 6-22.

9. Except where noted, arrays and strings must contain the exact number expected by the code (as indicated in the array or string description). If not, the code will eventually abort with a (hopefully) descriptive error message or messages.

10. New users reading these instructions for the first time and unfamiliar with the THREEDANT input may find it helpful to follow the sample input in Appendix A while reading this section.

11. Array names are shown here in upper case. What you should actually input for them will depend upon the code's implementation on your platform. At the present time, on most platforms, you should use lower case input.

12. Items in italics in the input instructions indicate actual values that may be entered for an array. You will frequently find switches where the input is the digit 0 or the digit 1. This will be represented by *0/1* in the input description. In other arrays where an exact character string is required such as "ISOTXS" in the LIB array, you will find the notation *ISOTXS*. Note that in this notation, the word is both upper case and italicized. This combination means you must enter exactly those characters. Again, although the characters will be shown here in upper case, what you should actually input for them will depend upon the code's implementation on your platform.

13. When a template for the input form is given, as for the MATLS array, the style in the template tells the user what is expected. If an input word or value is lower case and italicized, the user is to replace that position with the entry of his choice. If the input word is in italicized style and in upper case, the user is to input exactly those characters to achieve the desired result. Depending on the implementation on your platform, the input word, itself, is usually in lower case.

14. Units to be used for the input quantities are not spelled out as they only need to be self consistent. However, the following are commonly used: Dimensions in centimeters, isotopic cross sections in barns per atom; then it follows that atom densities are in atoms per barn-centimeter. Sources are particles per cm^3 per second for volumetric sources and particles per cm^2 per second for boundary sources; fluxes will then be in particles per cm^2 per second.

Title Line Details

Title Line Control (format 3I6)[a] {Required}

Word	Name	Comments
1	NHEAD	Number of title lines that follow.[b]
2	NOTTY	Suppress output to on-line user terminal? *0/1* = no/yes.
3	NOLIST	Suppress listing of all ASCII text input? *0/1* = no/yes. (default=no)

a. WARNING! Note that this first line is in fixed format.

b. Follow this control line with NHEAD title lines containing descriptive comments. Each title line may contain up to 72 characters.

Block-I Details: Dimensions and Controls

Dimensions {Required}

Name	Comments
IGEOM	Geometry. *14/15* = x-y-z/r-z-θ or use one of the following character strings: *X-Y-Z, R-Z-T*
NGROUP	Number of energy groups.
ISN	S_n order to be used. If ISN is negative, code will use the Chebychev-Legendre (IQUAD=2) quadrature set. (See IQUAD input on page 6-54 .)
NISO	Number of physical isotopes on the basic input cross-section library.
MT	Number of physical materials[a] to be created.
NZONE	Number of geometric zones[b] in problem.
IM	Number of coarse mesh intervals[c] in the x (or r) direction.
IT	Total number of fine mesh intervals[d] in the x (or r) direction.
JM	Number of coarse mesh intervals in the y (or z) direction.
JT	Total number of fine mesh intervals in the y (or z) direction.
KM	Number of coarse mesh intervals in the z (or θ) direction.
KT	Total number of fine mesh intervals in the z (or θ) direction.

a. Material is defined on page 6-44.

b. Zone is defined on page 7-13.

c. Coarse mesh is defined on page 7-13.

d. Fine mesh is defined on page 7-13.

Storage Requirements {Optional}

Name	Comments
MAXSCM	Length of SCM desired (default=40000_{10})
MAXLCM	Length of LCM desired (default=140000_{10})

Note: The above input (Dimensions plus Storage Requirements) for Block-I will cause the code to attempt to produce a full run, subject to availability of the input normally found in the other Blocks. The controls below allow shortened print files, partial runs (say, of only the input module), or cause the code to ignore any of the other input Blocks present. For full details on their use, see "PIECEWISE EXECUTION" on page 13-19.

Run Configuration Controls {Optional}

Name	Comments
NOSOLV	Suppress solver module execution. *0/1* = no/yes.
NOEDIT	Suppress edit module execution. *0/1* = no/yes.
NOGEOD	Suppress writing GEODST file even though the geometry input (Block-II) may be present. *0/1* = no/yes.
NOMIX	Suppress writing mixing files even though the mixing input in Block-IV may be present. *0/1* = no/yes.
NOASG	Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. *0/1* = no/yes.
NOMACR	Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. *0/1* = no/yes.
NOSLNP	Suppress writing the SOLINP file even though Block-V may be present. *0/1* = no/yes.
NOEDTT	Suppress writing the EDITIT file even though Block-VI may be present. *0/1* = no/yes.
NOADJM	Suppress writing the ADJMAC file even though an adjoint calculation is called for. *0/1* = no/yes.

Note: Default on all these controls is no.

Block-II Details: Geometry

Geometry Arrays[a] {Required}

Name	Comments
XMESH [IM+1]	x (or r) coordinates of coarse mesh edges.
YMESH [JM+1]	y (or z) coordinates of coarse mesh edges.
ZMESH [KM+1]	z (or θ) coordinates of coarse mesh edges. When r-z-θ geometry used, θ is in revolutions.
XINTS [IM]	Number of fine meshes in each coarse x (r) mesh
YINTS [JM]	Number of fine meshes in each coarse y (z) mesh
ZINTS [KM]	Number of fine meshes in each coarse z (θ) mesh
ZONES [IM;JM*KM]	Zone number[b] for each coarse mesh. This array defines the geometric zones to which cross-section materials are assigned via the ASSIGN input array of Block-IV. The zone number must not be greater than NZONE.

a. Definitions of coarse mesh, fine mesh, and zone are given in the chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 7-1. See note on units on page 6-31. The information entered in this block is written to the CCCC standard interface file GEODST.

b. A zone number of zero indicates the mesh contains a void, and no cross section will be associated with that mesh. The zero zone number is not counted in the total zone count NZONE.

Block-III Details: Nuclear Data

Nuclear Data Type and Options {Required}

Name	Comments
LIB	Name[a] and form of the cross-section data file. Enter as a data item one of the following words:

Word	Description
ISOTXS[b]	CCCC standard isotope ordered binary cross-section file.
XSLIB	ASCII text library supplied in a separate file named XSLIB.
ODNINP	ASCII text library follows after this block of input (after the T of Block-III).
GRUPXS[c]	CCCC standard group ordered cross-section file.
BXSLIB	Binary library supplied as a separate file named BXSLIB. [See "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 10-12.
MACRXS[d]	Use existing files named MACRXS for SOLVER module, SNXEDT for EDIT module. These files were created in a previous run. 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.
XSLIBB	See "XSLIBB Card-Image Library File" on page 10-12.
MACBCD	ASCII form of MACRXS file.
MENDF	(LANL only) See "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.
MENDFG	(LANL only) See "The Los Alamos MENDF5G Gamma Cross-Section Library" on page 10-14.
other	If a word other than those listed above is entered, the code will use the file with that word as its name, provided that file exists in the user's file space. Such a file must be structured as an XSLIB file.

Nuclear Data Type and Options (Cont.) {Required}

Name	Comments
WRITMXS {optional}	Controls the code's writing certain ASCII cross-section files.[c] Enter one of the following words: Word — Description *MACBCD* — Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file. *XSLIBB* — Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file. *XSLIBE* — Creates the cross-section file named XSLIBE, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBE is in Los Alamos 6E12 format (IFIDO=0). *XSLIBF* — Creates the cross-section file named XSLIBF, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBF is in FIDO fixed-field format (IFIDO=1).
LNG {optional}	Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.
BALXS {optional}	cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using! Value — Description *-1* — balance cross sections by adjusting absorption cross section. *0* — do not balance cross sections. (default) *1* — balance cross sections by adjusting self-scattering cross section.
NTICHI {optional}	MENDF fission fraction to be used for the problem (LANL only). *1/2/3* = Pu239/U235/U238 (default is U235). Will be overridden by any CHIVEC input described below or by any zone-dependent CHI in input Block-V.
CHIVEC [NGROUP] {optional}	Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.

a. On UNIX systems, the user may specify a search path for some of these files using the environment variable SNXSPATH. See "Library Search Path" on page 6-98 for details.

b. The CCCC standard for file ISOTXS does not allow the inclusion of the 2L+1 term in the higher order scattering cross section. However, if you have a nonstandard file which contains the 2L+1 term, you may override by setting I2LP1=1. See "Text Cross-Section Library Format" on page 6-41. THREEDANT will then convert the cross sections to the appropriate internal form.

c. The 2L+1 term on GRUPXS is treated the same as for ISOTXS. See footnote b.

d. In the convention used in this user's guide, a MACRXS library contains "material" cross sections; all the other libraries contain "isotope" cross sections.

e. See "COUPLED NEUTRON-GAMMA CROSS SECTIONS" on page 10-15.

Alternate Library Name {Optional}

Name	Comments
LIBNAME	Alternate name of the library file. May be used only with certain types of libraries. See Table 6.1.

The entries in the LIB input variable normally dictate both the form and the name of the cross section library. If the user specified ISOTXS, for example, the code would look for a file named ISOTXS and expect it to be in the CCCC format for an ISOTXS file.

For some libraries, the user may specify the form in the LIB array and specify separately the name in the LIBNAME array. The libraries that can be treated this way are shown inTable 6.1.

Table 6.1 LIBNAME Availability

LIB	LIBNAME AVAILABLE?
MACRXS	No
GRUPXS	Yes
ISOTXS	Yes
BXSLIB	Yes
ODNINP	No
MACBCD	No
XSLIBB	No
MENDF[a]	No
MENDFG[b]	No
XSLIB	Yes
other	Ignored

a. Available only at Los Alamos.

b. Available only at Los Alamos.

The BXSLIB file requires special treatment. It is normally created when the original library is a text library in the ODNINP or XSLIB form. In subsequent runs, this binary BXSLIB file may be used as the source of the cross-section data. The user may wish to save this file under another name. The program, in future runs, may then access the library for reading by using LIBNAME to specify that name.

This is a wise procedure because some cases using the BXSLIB form as input also require rewriting it in order to add new information. When this situation arises, the rewritten file is always named BXSLIB. Thus, if the original BXSLIB form library had a different name, it would be protected from being overwritten. For the remainder of the current run, the program will access the file named BXSLIB.

Text Cross-Section Library Format {Required if LIB= XSLIB or LIB=ODNINP}

Name	Comments
MAXORD	Highest Legendre order in the scattering tables.
IHM	Number of positions (entries) in each row of the cross-section table.
IHT	Position number of the total cross section.
IHS {optional}	Position number of the self-scatter cross section. (default = IHT + 1).
IFIDO {optional}	Format of the cross-section library. *-1/0/1/2* = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.
ITITL {optional}	A title line precedes each table. *0/1* = no/yes
I2LP1 {optional}	Higher order scattering cross sections on the library contain the 2L+1 term. *0/1* = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.
SAVBXS {optional}	Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. *0/1* = no/yes.
KWIKRD {optional}	Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? *0/1* = no/yes (default=yes).
NAMES [NISO] {optional}	Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).
EDNAME [IHT-3] {optional}	Character name for each of the EDIT cross-section positions used in the cross-section edits. These are the positions before the absorption cross section in the cross-section table. (default names are: EDIT1, EDIT2, . . .etc.).
NTPI [NISO] {optional}	Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).
VEL [NGROUP] {optional}	Speeds for each group. Needed only for alpha calculations.
EBOUND [NGROUP+1] {optional}	Energy group boundaries.

ASCII text libraries may be entered in one of the four forms indicated by the IFIDO input. All four forms share the following features: Cross sections are entered in a table optionally preceded by a title line. A table consists of NGROUP rows of entries. Each row contains the cross sections for a single group and consists of IHM entries. The user specifies the positions in the row occupied by the total and selfscattering cross sections. Order within a row (e.g., for group g) is then as follows:

$$\ldots \sigma_{abs},\ \nu\sigma_{f},\ \sigma_{total},\ \ldots \sigma_{g+2\rightarrow g},\ \sigma_{g+1\rightarrow g},\ \sigma_{g\rightarrow g},\ \sigma_{g-1\rightarrow g},\ \sigma_{g-2\rightarrow g},\ \text{etc.}$$

Notice that all terms in the scattering matrix are in positions relative to that of the self-scattering position and the rest of the cross sections are in positions relative to the position of the total cross section. The positions before the absorption cross section are frequently used for edit cross sections. For more detail, see "Ordering of Cross Sections Within a Cross-Section Table" on page 10-10.

Different Legendre orders are in different tables, which follow in order.

The user may order the group structure either by increasing energy or by decreasing energy. However, it is conventional and desirable for most problems to order it by decreasing energy, that is, group one is the highest energy. In that case, the scattering cross sections to the left of $\sigma_{g\rightarrow g}$ such as $\sigma_{g+1\rightarrow g}$ are upscattering terms and the terms to the right of $\sigma_{g\rightarrow g}$ are the downscattering terms.

In the Los Alamos format, the table is entered with a standard Fortran 6E12 format.

For greater precision in your input, use the 4E18 option.

In the fixed field FIDO format that ANISN[8] uses, entries are made in six twelve-column fields. Each twelve-column field is divided into three subfields, a two-column numeric field, a one-column character field, and a nine-column numeric field. See page 9-19 for details if you are not familiar with this input. The last field in each table must have the character T in the character position. No array identifier should be used. This format also restricts the usable input operators to T, *, R, -, +, and Z.

In the free field form, entries do not have to be in designated columns. Rather, the rules specified in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1 apply. Each table in this form is also terminated with the character T. No array identifier (i.e., array name with appended equals sign) should be used.

Block-IV Details: Cross-Section Mixing

A short summary of the primary mixing arrays, MATLS and ASSIGN, is given here for quick reference. Normally, THESE TWO ARRAYS ARE REQUIRED and, in most problems, would be the only arrays in this block. Other mixing arrays are also briefly described.

There are actually several nested levels of mixing. Each level has the job of calculating values from expressions of the form: $\Sigma_g = \sum_{i=1}^{k} N_i \sigma_{i,g}$ for each group, g. The user's job is to input the N_i for all the k components of the mixture and to specify each component, i. Component i has the cross section, $\sigma_{i,g}$. In common usage, for the first level of mixing, $\sigma_{i,g}$ is the effective microscopic cross section, and N_i is the atom density of isotope i, and Σ_g is then the macroscopic cross section of some material. In a higher level of mixing, these materials may be homogenized into a single material by using their volume fractions for the N_i. With several nested levels, the user has a great deal of flexibility in defining what Σ_g is for that level. A more complete discussion of mixing will be found in the chapter "MATERIAL MIXING TUTORIAL" starting on page 11-1.

A discussion of cross section processing is outside the scope of this document, but it should be noted that the user needs to be aware of the processing that is inherent in the input library. For instance, for materials in which there are isotopes with cross-section resonances, self shielding of the cross sections for these isotopes may be important and this effect must have been considered in the preparation of the "effective" microscopic cross sections for these isotopes. Since the self shielding is dependent on the amounts and types of the other isotopes in the material, the "effective" cross section is strictly valid only for use in a mixture which has the same composition as was used in the self shielding calculation. If the user desires to use this same "effective" microscopic cross section in some other composition (mix) of material, it is up to the user to verify the accuracy of this approach.

Primary Mixing Arrays {Required}

Name	Description
MATLS[a] [-;MT]	Instructions for mixing "isotopes" or premixes into "materials." See details below.
ASSIGN[b] [-;NZONE]	Assignments of materials to geometric zones. See below.
PREMIX [-;-] {optional}	Instructions for mixing "isotopes" into premixes. See below.

a. The information entered in the MATLS array is written to the CCCC standard interface files NDXSRF and ZNATDN.

b. Information entered in the ASSIGN array is written to the code-dependent interface file ASGMAT.

In order to understand how cross sections are mixed and the resultant material placed in the problem, we first need a little conceptual information.

The key entities used in specifying the cross-section spatial distribution are coarse mesh, zone, isotope, and material.

The basic geometry of the problem is defined with the coarse meshes specified in Block-II. The geometric areas called zones are also defined there using the ZONES array; the ZONES array designates the zone number assigned to each coarse mesh.

Here in Block-IV, we mix cross sections and assign them to the zones created in Block-II. For the purposes of this discussion, the cross sections found on the input library belong, by definition, to "isotopes," no matter what their true nature. These "isotopes" may then be mixed to form materials, using the MATLS array. Materials are then assigned to zones using the ASSIGN array.

MATLS input array

The general form of a MATLS mix instruction is shown below:

$$\text{MATLS=}\ \ mat_1\ \ comp_1\ den_1,\ \ comp_2\ den_2,\ \text{...etc....}\ ;$$

where mat_1 is the desired character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional materials (i.e., mat_2, mat_3, and so on up to the required number, MT) are defined in subsequent strings. Each string may contain as many components as necessary (actual limit = 500). A component is usually an isotope from the library, but may also be a temporary material created by the PREMIX array (see below).

When the component is an isotope, the den_i is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 6-49).

Short form: MATLS= *ISOS*

This form specifies that there should be as many materials as isotopes and that isotope number 1 is to be used for material number 1, isotope number 2 is to be used for material number 2, and so on.

In the special case where there is only a single component in a material and its density is unity, the density entry may be omitted as in the first material below:

MATLS= mat_1 $comp_1$; mat_2 $comp_2$ den_2; ...etc.... ;

ASSIGN input array

The general form of the ASSIGN instruction is shown below:

ASSIGN= $zone_1$ mat_1 vol_1, mat_2 vol_2, ...etc.... ;

where $zone_1$ is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). mat_1, mat_2, and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" vol_1, vol_2, and so on. Additional zones (i.e., $zone_2$, $zone_3$, and so on up to the required number, NZONE) are defined in subsequent strings. Although it is highly recommended that you use character names, here it is convenient to use the numeral for the zone name because it is the same numeral entered in the ZONES array.

Short form: ASSIGN= *MATLS*

This form specifies that there are as many zones as there are materials, and that material number 1 is to be assigned to zone number 1, material number 2 to zone number 2, and so on.

NOTE: The short form ASSIGN=*MATLS* can not be used if you intend to use the ASGMOD input array described later in this section.

PREMIX input array

The PREMIX array forms temporary materials in a way exactly analogous to the way that permanent materials are formed in the MATLS array. The difference in treatment is that the temporary materials created by PREMIX exist only long enough to complete the mixing; they are not available for assignment to geometric zones, nor are they available for use in material edits.

The general form of a PREMIX mix instruction is shown below:

PREMIX= $tmat_1$ $comp_1$ den_1, $comp_2$ den_2, ...etc.... ;

where $tmat_1$ is the character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional temporary materials (i.e., $tmat_2$, $tmat_3$, and so on) may be defined in subsequent strings. A component may be either an isotope from the library or another temporary material created by PREMIX.

The PREMIX array is useful for organizing the mixing input. For instance, it is frequently useful to mix the cross sections for a molecule of water and then in subsequent mix instructions, to input the molecular density of water as opposed to entering the atom density for both hydrogen and oxygen. Other examples are to form average cross sections for an element composed of many isotopes, or to form full density materials and then in later mix instructions to put in the volume fraction of the full density material.

Character Names vs. Numeric Names

In the foregoing discussion, isotopes, materials, and zones were identified by their character names. Optionally, they may be referred to by their ordinal number. Thus, 2 for an isotope name would call for the second isotope on the library. However, this practice is NOT recommended.

THE CHARACTER NAME FORM IS HIGHLY RECOMMENDED. It provides the most straightforward, most readable form. If the character name form is used, the naming input arrays in the following table are not needed.

Using the character name form in one array and the numeric name form in another array is particularly discouraged. However, should one wish to use the numeric form in the MATLS and/or ASSIGN arrays, and then subsequently associate character names with the ordinal numbers, one can use the naming arrays in the following table to do so. This situation could arise if, for some reason, one wanted to use material numbers in the MATLS array, but use character material names in the ASSIGN array.

When the library is of the MENDF form, the character names that must be used for the isotope names are discussed in "The Los Alamos MENDF5 Cross-Section Library" on page 10-13.

Mixing Array for a Concentration Search {Optional}

Name	Description
ASGMOD[a] [-;-]	C_1 parameters used in concentration searches. See the discussion below.

a. The information entered in the ASGMOD array is written to the ASGMAT file together with the information from the ASSIGN and CMOD arrays.

ASGMOD input array

The ASGMOD array is used in conjunction with the ASSIGN array when one wishes to vary the composition of a zone or zones in order to achieve a certain value of k-effective or alpha (i.e., in a concentration search). The concentration (or volume fraction) of material x in zone z is given by the following expression:

$$C(z,x) = C_0(z,x) + C_1(z,x)*CMOD$$

where $C_0(z,x)$ is the base concentration of material x in zone z. This is the concentration (or volume fraction) entered in the ASSIGN array for material x. In these arrays, x is not any kind of an index; correspondence is made by name, rather than by position within the array. Thus, for instance, in a problem that had ten materials, we might only assign one of them to a given zone. It would then probably be in the first position in the ASSIGN array string for that zone even though it might have been say, sixth in the list of all materials.

$C_1(z,x)$ is the corresponding entry in the ASGMOD array for material x in zone z.

CMOD is the search parameter (sometimes called search eigenvalue) that will be varied by THREEDANT in order to achieve the desired k-effective or alpha value. In a search calculation, the initial value for CMOD will be the input value EV.

The general form of the ASGMOD instruction is shown below:

ASGMOD= *zone* mat_m vol_m, mat_n vol_n, ...etc.... ;

where *zone* is the character name of any zone in the problem, mat_m, mat_n, and so on are the character names of any of the materials that will be present in this zone, and vol_m, vol_n, and so on are the C_1 values for respectively, mat_m, mat_n, and so on. Additional zones may be specified in subsequent strings. All zones do not have to appear in the ASGMOD array nor do all materials within a zone have to appear in the string for that zone.

Concentration Modifier {Optional}

Name	Description
CMOD	Concentration modifier. Input value is not used in a search. See the discussion below.

The concentration modifier, CMOD, is varied by THREEDANT during a search calculation. For any other type of calculation, a value of CMOD may be input and the composition of the zones will be calculated using the expression above for C(z,x).

Fine Mesh Mixing[a] {Optional}

Name	Description
FMMIX	Read the composition of each fine mesh from the file LNK3DNT. *0/1* = no/yes.

a. x-y-z geometry only.

The fine mesh mixing algorithm is designed for a general geometry option in x-y-z geometry using a volume fraction method on the fine mesh. It implies that one has an auxiliary code which will generate the volume fraction data from a general geometry description and store it on the file LNK3DNT.

Miscellaneous Mixing Input {Optional}

Name	Comments
MATNAM [MT]	Character material names for Materials. Used only if the mat_1 name used in the MATLS array was integer. First entry in MATNAM array is the desired character name for Material number 1, second entry is the desired character name for Material number 2, etc.
ZONNAM [NZONE]	Character zone names for Zones. Used only if the zone name entry in the ASSIGN or ASGMOD array was integer. First entry in the ZONNAM array is the desired character name for Zone number 1, second entry is the desired character name for Zone number 2, etc.
MATSPEC [≤MT]	Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions. Allowable entries are the words: *ATDENS* (default) atomic densities *ATFRAC*[a] atomic fractions *WTFRAC* weight fractions Can be input as a vector with up to MT entries (one for each Material) [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13.] If less than MT entries are made, the last entry will be used to fill out the array to a length of MT.
ATWT [≤2*NISO] {required[b]}	Atomic weights of the isotopes. If using MATSPEC=ATFRAC or WTFRAC, atomic weights must be available to the code. Entries for the ATWT array are made in pairs, as follows: ATWT= iso_1 $atwt_1$ iso_2 $atwt_2$ .... where iso_n is the isotope name (identifier) for isotope n on the cross-section library and $atwt_n$ is that isotope's atomic weight. [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 11-13].

a. ATFRAC and WTFRAC cannot be used with PREMIX.

b. Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.

Block-V Details: Solver Input

Desired Calculation {Required}

Name	Comments
IEVT	Calculation type: Enter one of the following values:. Value — Description *0* — source *1* — k_{eff} *2* — α (time absorption) search *3* — concentration search *4* — dimension search
ISCT	Legendre order of scattering (default = 0).
ITH	*0/1* = direct/adjoint calculation (default = 0).
IBL	Left boundary condition[a]: *0/1/3* = vacuum/reflective/white (default = vacuum).
IBR	Right boundary condition: *0/1/3* = vacuum/reflective/white (default = vacuum).
IBT	Top boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).
IBB	Bottom boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).
IBFRNT	Front boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).
IBBACK	Back boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).

a. The left boundary condition applies only for x-y-z geometry.

Iteration Controls {Required}

Name	Comments
EPSI	Convergence precision (default=0.0001).
IITL	Maximum number of inner iterations per group at first (default=1).
IITM	Maximum number of inners allowed when near fission source convergence (default chosen by code).
OITM	Maximum number of outer iterations (default=20).
ITLIM	Number of seconds time limit (default=unlimited).
NOSIGF[a]	Inhibit fission multiplication in a fixed source problem. *0/1* = no/yes.

a. The situation envisioned by this option is that a fission problem has previously been run in which case a FIXSRC file will have been automatically written. That FIXSRC file will contain the pointwise source given by $(1/k_{eff})\chi_g\Phi$ where Φ is the fission distribution. Then the problem is rerun with NOSIGF=1 and INSORS=1 to achieve the same answer as the original. This can then be used to study differences from the original system that do not impact the fission source.

K-Code Convergence {Optional}

Name	Comments
KCALC	Special Criticality Convergence Scheme. *0/1* = no/yes.

A special convergence scheme may be invoked for problems which require a good eigenvalue, but do not require tight convergence of the pointwise fluxes. It consists of converging the eigenvalue, but not the pointwise fluxes. Normally both must be converged. It also sets the default for eigenvalue convergence to 0.001 rather than 0.0001. To invoke this option to save running time, set the input parameter KCALC to unity.

Output Controls {Optional}

Name	Comments
FLUXP	Final flux print. *0/1/2* = no/isotropic/all moments.
XSECTP	Cross-section print. *0/1/2* = no/principal/all .
FISSRP	Fission source rate print. *0/1* = no/yes.
SOURCP	Source print. *0/1/2/3* = no/as input/normalized/both .
ANGP	Print angular flux. *0/1* = no/yes. CAUTION! This is very LARGE output. ANGP=1 will cause the RAFLXM or AAFLXM file to be written.
BALP	Coarse Mesh Interval Print Options. Enter one of the following values: Value — Description *0* — None *1* — Print coarse mesh balance tables.
RAFLUX	Prepare angular flux file (RAFLXM or AAFLXM). *0/1* = no/yes.
RMFLUX	Prepare flux moments file (RMFLUX or AMFLUX). *0/1* = no/yes.

Miscellaneous Solver Input {Optional}

Name	Comments
TRCOR	Apply transport correction[a] to cross sections on MACRXS file. Enter one of the following words: Word — Description *DIAG* — Use diagonal transport correction. *BHS* — Use Bell-Hansen-Sandmeier correction. *CESARO* — Use Cesaro "correction." *NO* — (or omit entry) Use no correction.
NORM	Normalize the fission source rate to this value when IEVT.GE.1 or normalize the inhomogeneous source rate to this value when IEVT.LT.1. NORM=0 means no normalization. (Integral of source rate over all angle, space, and energy = NORM, except for k_{eff} problems where the integral is equal to NORM*k_{eff}.) Any fluxes printed here (i.e., caused by setting FLUXP nonzero) will be normalized consistently with this source rate.
CHI [NGROUP;M]	Fission fraction born into each group.[b] Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.
DEN [IT;JT*KT] or	Density factor to use at each fine mesh point.
DENX [IT][c] and/or	Density factor to use at each fine x-mesh (default=1).
DENY [JT] and/or	Density factor to use at each fine y-mesh (default=1).
DENZ [KT]	Density factor to use at each fine z-mesh (default=1).
WDAMP [NGROUP]	Flags to activate adaptive weighted diamond differencing (AWDD)[9] for each group. *0.0*/W = no/activate AWDD with parameter W.[d] If W = 0.0, the default diamond with fixup is used.

a. For more information, see "Transport Corrections for the Cross Sections (TRCOR)" on page 7-31.

b. This input will override any previous CHI from earlier blocks or from any cross-section library which contains CHI.

c. In this second form, the density factor DEN(i,j,k), at mesh interval (i,j,k) is computed as follows:

$$DEN(i,j,k) = DENX(i)*DENY(j)*DENZ(k)$$

d. Recommend $1.0 < W < 4.0$ for shielding applications.

Quadrature Details

Name	Description
GRPSN [NGROUP][a]	S_n order to be used for each group.
IQUAD	Source of the quadrature constants. Enter one of the following values: *-3* Get constants from SNCONS file. *-2* Triangular Chebychev-Legendre built-in set. Any even value for ISN can be used up to 50. From 50 to 100, ISN must be in multiples of 10. See Ref. 7 for details. *1* Traditional built-in constants. Any even value for ISN can be used between 2 and 16, inclusive. (This is the default). *2* Rectangular Chebychev-Legendre built-in set. (REQUIRES ISN negative in Block-I!) Any even value for the absolute value of ISN can be used up to 50. From 50 to 100, the absolute value of ISN must be in multiples of 10. See Ref. 7 for details.
WGT[b] [MM[c]] {optional}	Quadrature weights.
MU [MM] {optional}	Mu cosines.
ETA [MM] {optional}	Eta cosines.

a. Value must be less than or equal to abs(ISN) in Block-I.

b. Presence of the WGT, MU, and ETA arrays overrides the IQUAD input.

c. MM=ISN*(ISN+2)/8 for iquad = -2,1.
MM = (ISN/2)**2 for iquad = 2.

Flux Guess From a File {Optional}

Name	Comments
INFLUX	Read the initial flux guess from a file.[a] *0/1* = no/yes. If ITH=0 and ISCT>0, and the flux moments file RMFLUX exists, read the initial flux guess from RMFLUX. Otherwise, read the initial flux guess from the RTFLUX file. If ITH=1 and ISCT>0, and the adjoint moments file AMFLUX exists, read the initial flux guess from AMFLUX. Otherwise, read the initial flux guess from the ATFLUX file.

a. There is presently no text input flux guess available for THREEDANT.

General Eigenvalue Search Control[a] { IEVT >1}

Name	Comments
IPVT	Type of eigenvalue to search for in a concentration or dimension search. *0/1/2* = none / k_{eff} / α. (default = 1).
PV	Value of k_{eff} or α to which to search. (default = 1.0 if IPVT=1, 0.0 if IPVT=2.
EV	Initial search parameter. Value at which to start the search parameter. (default=0).
EVM	Initial search parameter increment. Amount by which to change search parameter in the first step of a search. (REQUIRED - there is no default).
XLAL	Lambda lower limit for search. (default = 0.01).
XLAH	Lambda upper limit for search. (default = 0.5).
XLAX	Lambda convergence criterion for second and subsequent search steps. (default = 10*EPSI).
POD	Parameter oscillation damper. (default=1.0).

a. See "Eigenvalue Searches" on page 7-33 for definitions of these quantities.

THREEDANT can vary the composition or dimensions of a zone (or zones) in order to achieve a desired k-effective or alpha value. The search input consists of the above general search input plus input specific to the type of search being performed.

Dimension Search Input {Required if IEVT=4}

Name	Comments
XM [IM]	x-dimension fractional change per coarse mesh.
YM [JM]	y-dimension fractional change per coarse mesh.
ZM [KM]	z-dimension fractional change per coarse mesh.

The dimension search requires the XM and/or YM and/or ZM input as well as the general search input above. During the search, THREEDANT varies the search parameter (sometimes called the search eigenvalue) denoted by EV in the following expressions to change the coarse mesh boundaries:

$$XMESH_{i+1} = XMESH_i + \{XMESH_{i+1}\text{-}XMESH_i\} * [1.0+EV*XM_i], \quad i=1,\ldots,IM$$

$$YMESH_{j+1} = YMESH_j + \{YMESH_{j+1}\text{-}YMESH_j\} * [1.0+EV*YM_j], \quad j=1,\ldots,JM$$

$$ZMESH_{k+1} = ZMESH_k + \{ZMESH_{k+1}\text{-}ZMESH_k\} * [1.0+EV*ZM_k], \quad k=1,\ldots,KM$$

Although they may seem a bit awkward at first, the user will find these expressions to be quite flexible. With proper choice of the XM_i, YM_j, and ZM_k values, the user can move any or all of the coarse mesh boundaries while allowing others to remain stationary. The quantities in { } in the above expressions are always formed from the original input values.

Concentration Search Input {Required if IEVT=3}

Name	Description
The solver input for a concentration search is to set IEVT = 3 (page 6-50) and input the general eigenvalue search controls. But you must also input the ASGMOD[a] array in Block-IV.	

a. A concentration search involves the mixing instructions. A discussion of the ASGMOD array is found in the mixing input description on page 6-47.

Volumetric Source Options {Optional}

Name	Comments
INSORS	Read source from interface file FIXSRC. 0/1 = no/yes
----- For a text-input source, choose one of the following options:	
Option 1:	
SOURCE [NGROUP; NMQ]	Source spectrum for each of NMQ[a] moments. (Spatial distribution is assumed to be flat with value unity)
Option 2:	(input all three arrays)
SOURCX [IT;NMQ][b]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z) spatial distribution for each moment.
SOURCZ [KT;NMQ]	z (or θ) spatial distribution for each moment.
(Spectrum is assumed to be flat with value unity)	
Option 3:	(input all four arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCX [IT;NMQ]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z) spatial distribution for each moment.
SOURCZ [KT;NMQ]	z (or θ) spatial distribution for each moment.
Option 4:	
SOURCF [IT;JT*KT*NGROUP*NMQ]	Spatial distribution for each row, group, and moment.
Option 5:	(input both arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCF [IT; JT*KT*NMQ]	Spatial distribution for each row and moment.

a. NMQ is not an input value but is computed from the number of strings read. NMQ must correspond exactly to the number of moments in a P_n expansion of the source. The number of moments is $(n+1)^2$. n must be less than or equal to ISCT. See page 12-24 for more details.

b. Only in option 4 is the complete pointwise source array, SOURCF(i,j,k,g,m), given. In all other cases, it must be formed from the lower dimension arrays that are input. That calculation is done by forming the product of those arrays. Thus, in option 3, where the source spectrum, SOURCE(g,m), and the spatial distributions SOURCX(i,m), SOURCY(j,m), SOURCZ(k,m) are given (for moment m), the full source at mesh point (i,j,k) in group g for moment m is calculated as follows:

SOURCF(i,j,k,g,m) = SOURCE(g,m)*SOURCX(i,m)*SOURCY(j,m)*SOURCZ(k,m)

Boundary Source Input {Optional}

Name	Comments
----- For a text-input source, choose one of the following options:	
Option 1: Isotropic Boundary Source	
SILEFT [NGROUP;JT*KT]	Isotropic source on the left face. (Spectrum at each y,z mesh interval.)
SIRITE [NGROUP;JT*KT]	Isotropic source on the right face.
SIBOTT [NGROUP;IT*KT]	Isotropic source on the bottom face.
SITOP [NGROUP;IT*KT]	Isotropic source on the top face.
SIFRNT [NGROUP;IT*JT]	Isotropic source on the front face.
SIBACK [NGROUP;IT*JT]	Isotropic source on the back face.
Option 2: Full Angular Boundary Source[a]	
SALEFT [MM[b]*4;NGROUP*JT*KT]	Angular flux on the left for each angle, group, and y,z mesh interval.
SARITE [MM*4;NGROUP*JT*KT]	Angular fluxes on the right face.
SABOTT [MM*4;NGROUP*IT*KT]	Angular fluxes on the bottom face.
SATOP [MM*4;NGROUP*IT*KT]	Angular fluxes on the top face.
SAFRNT [MM*4;NGROUP*IT*JT]	Angular fluxes on the front face
SABACK [MM*4;NGROUP*IT*JT]	Angular fluxes on the back face

a. The order of the angles is identical to that used in the S_n Constants table in the output file. The order of the angular octants is: mu < 0, eta < 0, tsi < 0; mu > 0, eta < 0, tsi < 0; mu < 0, eta > 0, tsi < 0; mu > 0, eta > 0, tsi < 0; mu < 0, eta < 0, tsi > 0; mu > 0, eta < 0, tsi > 0; mu < 0, eta > 0, tsi > 0, and mu > 0, eta > 0, tsi > 0, where each angular boundary source requires four octants for specification.

b. See "Quadrature Details" on page 6-54 for value of MM.

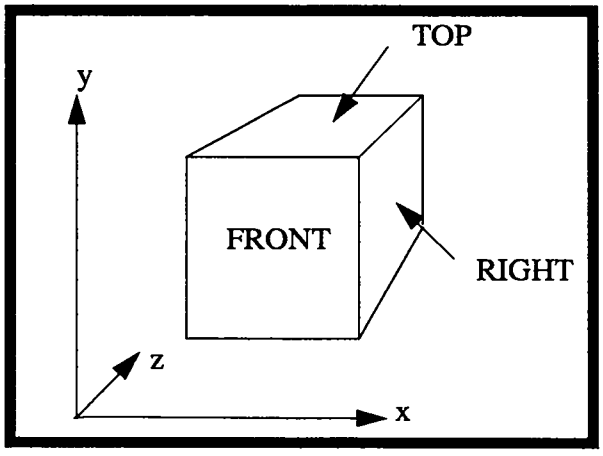


Figure 6.2 Orientation of Faces

Note the non-standard orientation of the z axis and accordingly that top is NOT in the z direction.

Within option 3 are four suboptions that involve different combinations of 'vector' type input for the boundary sources. All the vector input is defaulted to unity if not entered explicitly. The full angular source for the left face, for example, is constructed as follows for the 4 options (the source construction on the other faces is analogous):

Option 3a- S(m,g,y,z) = BSLFTG(g)*BSLFTY(y)*BSLFTZ(z)*BSLFTA(m)

Option 3b- S(m,g,y,z) = BSLFTG(g)*BSLFTYZ(y,z)*BSLFTA(m)

Option 3c- S(m,g,y,z) = BSLFTG(g)*BSLFTYZ(y,z,m)

Option 3d- S(m,g,y,z) = BSLFTG(g)*SALEFT(m,y,z)

Boundary Source Vector Input Combinations {Optional}

Left	Right	Bottom	Top	Front	Back
Option 3a: Boundary Source By Product Of Vectors					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
BSLFTY [JT]	BSRITY [JT]	BSBOTX [IT]	BSTOPX [IT]	BSFRNX [IT]	BSBAKX [IT]
BSLFTZ [KT]	BSRITZ [KT]	BSBOTZ [KT]	BSTOPZ [KT]	BSFRNY [JT]	BSBAKY [JT]
BSLFTA [MM*4]	BSRITA [MM*4]	BSBOTA [MM*4]	BSTOPA [MM*4]	BSFRNA [MM*4]	BSBAKA [MM*4]
Option 3b: Boundary Source By Product Of A Spectrum Vector, An Angular Distribution Vector, And A 2d Spatial Array					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
BSLFTYZ [JT;KT]	BSRITYZ [JT;KT]	BSBOTXZ [IT;KT]	BSTOPXZ [IT;KT]	BSFRNXY [IT;JT]	BSBAKXY [IT;JT]
BSLFTA [MM*4]	BSRITA [MM*4]	BSBOTA [MM*4]	BSTOPA [MM*4]	BSFRNA [MM*4]	BSBAKA [MM*4]
Option 3c: Boundary Source By Product Of A Spectrum Vector And A 3d Space-Angle Array					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
BSLFTYZ [JT;KT* MM*4]	BSRITYZ [JT;KT* MM*4]	BSBOTXZ [IT;KT* MM*4]	BSTOPXZ [IT;KT* MM*4]	BSFRNXY [IT;JT* MM*4]	BSBAKXY [IT;JT* MM*4]
Option 3d: Boundary Source By Product Of A Spectrum Vector And A 3d Angle-Space Array					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
SALEFT [MM*4; JT*KT]	SARITT [MM*4; JT*KT]	SABOTT [MM*4; IT*KT]	SATOPT [MM*4; IT*KT]	SAFRNT [MM*4; IT*JT]	SABAKT [MM*4; IT*JT]

Block-VI Details: Edit Input*

Edit Spatial Specifications {Required[a]}

Name	Comments
PTED	Do edits by fine mesh. *0/1* = no/yes.
ZNED	Do edits by zone. *0/1* = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below.)
POINTS [≤IT*JT*KT] {optional}	Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points)
EDZONE [IT;JT*KT] {optional}	Edit zone number for each fine mesh interval. USED ONLY IF ZNED=1. (default= SOLVER coarse mesh interval numbers, see ZONES array, Block-II on page 6-36)

a. Either PTED or ZNED or both must be unity in order to produce reaction rate edits.

* More details for the input for edits are given in chapter "RUNNING THE EDIT MODULE" starting on page 8-1.

Reaction Rates from Cross Sections[a] {Optional[b]}

Name	Comments
EDXS [≤NEDT] {required[c]}	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 the character name of the cross-section type. See the table"Edit Cross-Section Types by Position and Name" on page 6-64 or "MENDF Library Edit Cross Sections" on page 6-71 for the available names. NEDT is the total number of edit cross-section types available from the input cross-section library. (default = all shown in the table) Note: The cross-section types specified in this array apply to any or all of the following edit forms: RESDNT, EDISOS, EDCONS, EDMATS.
RESDNT	Do edits using the resident macroscopic cross section at each point. *0/1* = no/yes.
EDISOS [≤NISO]	Character names of the isotopes to be used in forming Isotopic reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDCONS [≤NISO]	Character names of the isotopes to be used in forming resident Constituent (partial macroscopic) reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDMATS [≤MT]	Character names of materials to be used in forming Material (macroscopic) reaction rates. The ordinal number may alternately be used, but is not recommended. (default = none).
XDF[d] [IT] YDF [JT] ZDF [KT]	Fine mesh density factors for the x(or r), y(or z) and z(or θ) directions, respectively. The density factor is used to multiply resident Constituent (see EDCONS), resident macroscopic (see EDMATS), and resident macroscopic (see RESDNT) reaction rates only. (Default= all values unity).

a. See chapter "RUNNING THE EDIT MODULE" starting on page 8-1 for further discussion.

b. But either something in this grouping or the next must be input in order to produce reaction rate edits.

c. You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.

d. If density factors were used in SOLVER to modify the cross sections at each mesh interval, the same density factors must be provided here in the XDF and/or YDF and/or ZDF arrays as well. The density factor at mesh interval (i,j,k) is computed as:

XDF(i)*YDF(j)*ZDF(k)

Edit Cross-Section Types by Position and Name

CROSS-SECTION INPUT VIA ISOTXS or GRUPXS			CROSS-SECTION INPUT VIA ASCII TEXT		
Type	EDIT Position	Name[a]	Type	EDIT Position	Name
chi	1	CHI.....	not used	1	CHI.....
nu-fission	2	NUSIGF..	nu-fission	2	NUSIGF..
total	3	TOTAL...	total	3	TOTAL...
absorption	4	ABS.....	absorption	4	ABS.....
(n,p)	5	N-PROT..	1[b]	5	EDIT1...[c]
(n,d)	6	N-DEUT..	2	6	EDIT2...
(n,t)	7	N-TRIT..	3	7	EDIT3...
(n,alpha)	8	N-ALPH..	.	.	.
(n,2n)	9	N-2N....	.	.	.
(n,gamma)	10	N-GAMM..	.	.	.
fission	11	N-FISS..	N=IHT-3	4+N	EDITN...
transport	12	TRNSPT..			

a. Names are eight characters. A period within a name in this table denotes a blank.

b. Denotes position in the cross-section table. All cross sections in 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 EDNAME array of Block-III.

Reaction Rates from User Response Functions {Optional[a]}

Name	Comments
RSFE [NGROUP;M] {required if user input response functions are desired}.	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 with NGROUP entries beginning with group 1.
RSFX [IT;M][b]	Response function x (or r) distribution for M functions.
RSFY [JT;M]	Response function y (or z) distribution for M functions.
RSFZ [KT;M]	Response function z (or θ) distribution for M functions.
{optional}	The above data are entered as M strings of IT, JT or KT entries beginning with mesh point 1. (default=1.0)
RSFNAM [M] {optional}	Character names for the user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM)

a. But either something from this grouping or the previous one must be input in order to produce reaction rate edits.

b. The m-th response function at space point (i,j,k) and energy group g is computed as:

RSFX(i,m) * RSFY(j,m) * RSFZ(k,m) * RSFE(g,m)

Energy Group Collapse Specifications {Optional}

Name	Comments
ICOLL [NBG]	Edit energy group collapsing option: Number of SOLVER energy groups in each EDIT broad group. The NBG entries must sum to NGROUP. (default = 1 energy group per EDIT broad group)
IGRPED	Print option on energy groups. Enter one of the following values: Value — Description *0* — Print energy group totals only *1* — Print broad groups only *2* — Print broad groups only (same as 1) *3* — Print both broad groups and totals

Reaction Rate Summing {Optional}

Name	Comments
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 "Edit Cross-Section Types by Position and Name" on page 6-64). 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. (for more detail, see "Response Function Summing Options" on page 8-13).
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 page 8-13 for more detail.

Mass Inventories {Optional}

Name	Comments
MASSED	Calculate and print mass inventories by zone. *0/1/2/3* = none/ solver zones/edit zones/both (default=1). This option is active only if atomic weights are present. See ATWT on page 6-49.

Power Normalization {Optional}

Name	Comments
POWER {required}	Normalize to POWER megawatts.[a] All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be normalized. Fluxes are normally not printed here in the EDIT module, although they may be extracted by using a unit response function. Any such fluxes will also be normalized to POWER. Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 6-53).
MEVPER {required}	MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power. For the power calculation, THREEDANT needs to know which cross section is the fission cross section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or GRUPXS library that designation is automatically provided (See "Edit Cross-Section Types by Position and Name" on page 6-64). But if one uses an ASCII text library, either ODNINP or XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 6-41).

a. Note that this normalization is meaningless if you are using the results of an adjoint run.

Miscellaneous Edit Items {Optional}

Name	Comments
RZFLUX	Write the CCCC standard zone[a] flux file RZFLUX or AZFLUX. *0/1* = no/yes.
RZMFLX	Write the code-dependent zone[b] flux moments file RZMFLX or AZMFLX. *0/1* = no/yes.
EDOUTF[c]	ASCII output files control. Enter one of the following values: Value — Description *-3* — Write both EDTOGX (without scalar fluxes) and EDTOUT files. *-2* — Write EDTOGX file (without scalar fluxes). *0* — Write neither file. (default) *1* — Write EDTOUT file. *2* — Write EDTOGX file (with scalar fluxes). *3* — Write both EDTOGX (with scalar fluxes) and EDTOUT files.
BYVOLP	Printed point reaction rates will have been multiplied by the mesh volume. *0/1* = no/yes.
AJED[d]	Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. *0/1* = regular/adjoint.
FLUXONE	Flux override. *0/1* = no/yes. Replaces all the input fluxes by unity. Useful for seeing the cross sections used in cross-section edits. WARNING! Meaningful reaction rates cannot be obtained when this switch is on.

a. RZFLUX and AZFLUX are organized by solver zones.

b. RZMFLX and AZMFLX are organized by solver zones.

c. See "ASCII File Output Capabilities (the EDOUTF Parameter)" on page 8-15.

d. See "Adjoint Edits" on page 8-15.

Special Plot Linkage {Optional}

Name	Comments
PRPLTED	Write an ASCII file of the pointwise reaction rates to link to the TECPLOT© plotting package available commercially for a SUN workstation. *0/1/2/3* = print only/nothing/tecplot file/both print and tecplot file.
IPLANE [≤IT]	x mesh numbers at which to write a y-z distribution.
JPLANE [≤JT]	y mesh numbers at which to write a x-z distribution.
KPLANE [≤KT]	z mesh numbers at which to write a x-y distribution.

To exercise this option, the user must have set PTED=1. The code will calculate reaction rates at all the fine mesh intervals, and any POINTS input will be ignored.

To link to the TECPLOT© code, the user chooses option 2 or 3. Separate ASCII files called rsp.dat and med.dat will be written for the response function and material edits, respectively. These files are in input form for the TECPLOT© preprocessor.

If option 0 (print only) is chosen, no TECPLOT© files will be written but the reaction rates will be printed. The format of this printout is organized in a two-dimensional way unlike the normal printout from the EDIT module.

MENDF Library Edit Cross Sections

Reaction Type	Name	Description
χ	CHI	fission spectrum
$\nu\sigma_f$	NUSIGF	effective nu-sigma-fission
σ_t	TOTAL	Total cross section
σ_a	ABS	absorption [a]
(n,n)	MEND1	elastic scattering
(n,n´)	MEND2	inelastic scattering
(n,2n)	MEND3	n,2n scattering
(n,3n)	MEND4	n,3n scattering
(n,γ)	MEND5	gamma production
(n,α)	MEND6	alpha production
(n,p)	MEND7	proton production
(n,f)	MEND8	direct fission
(n,n´)f	MEND9	second-chance fission
(n,2n)f	MEND10	third-chance fission
(n,F)	N-FISS	[(n,F) = (n,f) + (n,n´)f + (n,2n)f]
χ_p	MEND12	prompt fission spectrum (only for fissionable materials)
χ_t	MEND13	total fission spectrum (only for fissionable materials)

a. σ_a for group g is defined as $\sigma_a = \sigma_t - \sum_{g'} \sigma_{g \to g'}$

When using the Los Alamos MENDF5 cross-section library with the codes there are numerous edit cross sections available for use in the Edit Module. Since these come from the MENDF file, they are called upon with special character names in the Edit Module as part of the EDXS input. These names are defined above.

REFERENCES

1. 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.
2. 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.
3. R. D. O'Dell, F. W. Brinkley, D. R. Marr, and R. E. Alcouffe, "Revised User's Manual for ONEDANT: A Code Package for One- Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory report LA-9184-M, Revised, (December 1989).
4. 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).
5. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," Nucl. Sci. Eng. 64, 344 (1977).
6. R. E. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Multigroup Diffusion Equation," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 340-351.
7. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).
8. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693 (March 1967).
9. R. E. Alcouffe, "An Adaptive Weighted Diamond Differencing Method for Three-Dimensional XYZ Geometry," *Trans Am Nuc Soc.* **68**, Part A, 206 (1993).

APPENDIX A: SAMPLE INPUT

Sample Problem: Standard k_{eff} Calculation

A small but complete sample problem input is shown below. It is a four group calculation of the eigenvalue and eigenfunction of an X-Y-Z model of a sodium cooled fast reactor experiment. The geometric model contains five zones, core, radial and axial blankets, and control rods. Macroscopic P_0 cross sections for each material zone are entered in the input stream after Block-III as macroscopic cross sections where the total is in position 4 and the inscatter is in position 5.

In the edit input, the code is asked to calculate point reaction rates and reaction rate totals for each edit zone. These are based on the response functions as a function of group entered in RSFE which obtains the scalar flux for each group.

Sample Problem: Output Description

Selected items in the output listing are described here. We focus on items particular to a three-dimensional calculation. For a more thorough description of output items common to one-, two-, and three-dimensional calculations, the reader is referred to Appendix A in the chapter "ONEDANT USER'S GUIDE".

Block-I shows this to be a 14x14x30 fine mesh problem in 8x8x4 coarse meshes; and we are to do a 4 group, S8 calculation. In Block-II is given the mesh and zone specifications. A zone map for each coarse mesh k-plane is shown on page 6-84, and the mesh information is summarized on page 6-85. These edits are useful for verifying that the input geometry matches what was intended. Note: In this particular run, we are not using material 4. In Block-III, we have input the macroscopic cross sections for the 5 materials of the problem in 4 energy groups. Thus in Block-IV, the mixing instructions are the simple option. The cross sections as used for the calculation are printed starting on page 6-86. In Block-V, we specify that this is a k_{eff} calculation with reflecting boundaries on the left and bottom of the system and vacuum on all other boundaries. We also specify a convergence criterion of $5x10^{-5}$, and we provide the CHI values (fission spectrum) in this block. The rest of the input in this block specifies that we are to print the cross sections and gives the upper limit to the number of outers as 11. In Block-VI we indicate that we want point and zone edits for the response function RSFE only. We specify the edit zones for the zone edit by EDZONE which is on the fine mesh. The points we want values for are given by the POINTS array and are translated on page 6-90 for verification.

After the display of the code's interpretation of the input data, the actual calculational iteration monitor is shown on page 6-88. This monitor displays the convergence of the eigenvalue and fission source distribution as a function of outer iteration assuming the default iteration strategy of one inner iteration per group per outer until the fission source has nearly converged. In this example these are the first 6 outers; on the seventh outer, the scalar flux is iterated to convergence to give the final eigenvalue and fission source distribution. This problem converged to $5x10^{-5}$ in 29 total inners and in 12.2 CPU seconds on a Cray YMP. On page 6-89 is presented the integrated balance table by

group which gives an indication of goodness of solution as well as displaying important physical quantities. The output from the edit module begins on page 6-92 which is a zone and point edit of the scalar flux as was intended according to the input specifications. Note if further edits were desired, the edit input of Block-VI could be changed accordingly and the NOSOLV=1 set in Block-I. This would run only the edit module using the previously generated flux and geometry files.

Sample Problem: Output Listing

```
*****************************************************************************************************************************
*****************************************************************************************************************************
*
*            generalized input module run on 01/17/95 with solver version 10-13-94beta---- release 2.5      machine e
*
*****************************************************************************************************************************
*                  ***************************************************************************************
*                  *                                                                                     *
*                  *                    ...listing of cards in the input stream...                       *
*                  *                                                                                     *
*                  *    1.     2          0                                                              *
*                  *    2.  3D model of a small FBR.                                                     *
*                  *    3. Cross sections from input...                                                  *
*                  *    4. /                                                                             *
*                  *    5. / Block I...                                                                  *
*                  *    6. /                                                                             *
*                  *    7. igeom=x-y-z ngroup=4 isn=8 niso=5 mt=5 nzone=5 im=8 it=14 jm=8 jt=14          *
*                  *    8.   km=4 kt=30                                                                  *
*                  *    9.   maxscm=300000 maxlcm=600000                                                 *
*                  *   10.   t                                                                           *
*                  *   11. /                                                                             *
*                  *   12. / Block II...                                                                 *
*                  *   13. /                                                                             *
*                  *   14.  xmesh=0.0 15.0 30.0 35.0 40.0 45.0 50.0 55.0 70.0                            *
*                  *   15.       xints=3 3 1 1 1 1 1 3                                                   *
*                  *   16.  ymesh=0.0 5.0 15.0 30.0 40.0 45.0 50.0 55.0 70.0                             *
*                  *   17.       yints=1 2 3 2 1 1 1 3                                                   *
*                  *   18.  zmesh=0.0 20.0 75.0 130.0 150.0                                              *
*                  *   19.       zints=4 11 11 4                                                         *
*                  *   20. /                                                                             *
*                  *   21.  zones=3r2 2r3 2r2 5; 7r2 5; 6r2 2r5; 5r2 3r5; 4r2 4r5; 2r2 6r5;              *
*                  *   22.                2 7r5; 8r5;                                                    *
*                  *   23.         3r1 2r3 2r1 5; 7r1 5; 6r1 2r5; 5r1 3r5; 4r1 4r5; 2r1 6r5;             *
*                  *   24.                1 7r5; 8r5;                                                    *
*                  *   25.         3r1 2r3 2r1 5; 7r1 5; 6r1 2r5; 5r1 3r5; 4r1 4r5; 2r1 6r5;             *
*                  *   26.                1 7r5; 8r5;                                                    *
*                  *   27.         3r2 2r3 2r2 5; 7r2 5; 6r2 2r5; 5r2 3r5; 4r2 4r5; 2r2 6r5;             *
*                  *   28.                2 7r5; 8r5;                                                    *
*                  *   29.   t                                                                           *
*                  *   30. /                                                                             *
*                  *   31. / Block III...                                                                *
*                  *   32. /                                                                             *
*                  *   33. lib=odninp                                                                    *
*                  *   34.  ihm=8 iht=4 ihs=5 ititl=1 ifido=2  t                                         *
*                  *   35. core                                                                          *
*                  *   36. 1. 0.00745551 0.0206063 0.114568 0.0704326 3r0.0 /                            *
*                  *   37. 1. 0.0035254 0.00610571 0.205177 0.195443 0.0347967 2r0.0 /                   *
*                  *   38. 1. 0.00780136 0.00691403 0.329381 0.320586 0.00620863 0.00188282 0.0 /        *
*                  *   39. 1. 0.0274496 0.0260689 0.38981 0.36236 9.92975e-4 7.07208e-7 0.0  t           *
*                  *   40. axblkt                                                                        *
*                  *   41. 1. 0.00535418 0.013177 0.116493 0.0716044 3r0.0 /                             *
*                  *   42. 1. 0.00148604 1.26026e-4 0.220521 0.210436 0.037317 0.0 0.0 /                 *
*                  *   43. 1. 0.005353 1.5238e-4 0.344544 0.337506 0.00859855 0.00221707 0.0 /           *
*                  *   44. 1. 0.0134694 7.87302e-4 0.388356 0.374886 0.0016853 6.68299e-7 0.0 t          *
*                  *   45. nacrod                                                                        *
*                  *   46. 1.  3.10744e-4 0.0 0.0658979 0.0474407 3r0.0 /                                *
*                  *   47. 1.  1.13062e-4 0.0 0.10981 0.106142 0.0176894 0.0 0.0 /                       *
*                  *   48. 1.  4.48988e-4 0.0 0.186765 0.185304 0.00355466 4.57012e-4 0.0 /              *
*                  *   49. 1.  0.00107518 0.0 0.209933 0.208858 0.0010128 1.77599e-7 0.0 t               *
*                  *   50. cntrod                                                                        *
*                  *   51. 1. 0.00597638 0.0 0.184333 0.134373 3r0.0 /                                   *
*                  *   52. 1. 0.0176941 0.0 0.366121 0.318582 0.0437775 0.0 0.0 /                        *
*                  *   53. 1. 0.0882741 0.0 0.615527 0.519591 0.0298432 2.06054e-4 0.0 /                 *
*                  *   54. 1. 0.476591 0.0 1.09486 0.618265 0.00766209 8.71188e-7 0.0 t                  *
*                  *   55. radblk                                                                        *
*                  *   56. 1. 0.00743283 0.0189496 0.119648 0.0691158 3r0.0 /                            *
*                  *   57. 1. 0.0019906  1.75265e-4 0.242195 0.230626 0.0404132 0.0 0.0 /                *
*                  *   58. 1. 0.00679036 2.06978e-4 0.356476 0.348414 0.00957027 0.00268621 0.0 /        *
*                  *   59. 1. 0.0158015 0.00113451 0.379433 0.363631 0.00127195 1.99571e-7 0.0 t         *
*                  *   60.                                                                               *
*                  *   61. /                                                                             *
*                  *   62. / Block IV...                                                                 *
*                  *   63. /                                                                             *
*                  *   64.  matls=isos                                                                   *
*                  *   65.  assign=matls                                                                 *
*                  *   66.   t                                                                           *
*                  *   67. /                                                                             *
*                  *   68. / Block V...                                                                  *
*                  *   69. /                                                                             *
*                  *   70. ievt=1 ibl=1 ibb=1 ibfrmt=0 norm=1 iitl=1 oitm=11 xsectp=2                    *
*                  *   71.  epsi=5.0e-5 chi=0.583319 0.40545 0.011231 0.0                                *
*                  *   72.   t                                                                           *
*                  *   73. /                                                                             *
*                  *   74. / Block VI...                                                                 *
*                  *   75. /                                                                             *
*                  *   76.   pted=1 zned=1 edoutf=1                                                      *
*                  *   77.  edzone=7r2 2r3 2r2 3r5; 11r2 3r5; 1y 1; 10r2 4r5; 2y 1; 9r2 5r5; 1y 1;       *
*                  *   78.         8r2 6r5; 6r2 8r5; 3r2 11r5; 14r5; 2y 1; 3y 14;                        *
*                  *   79.                                                                               *
*                  *   80.         7r1 2r3 2r1 3r5; 11r1 3r5; 1y 1; 10r1 4r5; 2y 1; 9r1 5r5; 1y 1;       *
*                  *   81.         8r1 6r5; 6r1 8r5; 3r1 11r5; 14r5; 2y 1; 10y 14;                       *
*                  *   82.                                                                               *
*                  *   83.         7r1 2r4 2r1 3r5; 11r1 3r5; 1y 1; 10r1 4r5; 2y 1; 9r1 5r5; 1y 1;       *
*                  *   84.         8r1 6r5; 6r1 8r5; 3r1 11r5; 14r5; 2y 1; 10y 14;                       *
*                  *   85.                                                                               *
*                  *   86.         7r2 2r4 2r2 3r5; 11r2 3r5; 1y 1; 10r2 4r5; 2y 1; 9r2 5r5; 1y 1;       *
*                  *   87.         8r2 6r5; 6r2 8r5; 3r2 11r5; 14r5; 2y 1; 3y 14;                        *
*                  *   88.                                                                               *
*                  *   89. points=2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 / *
*                  *   90.       2759 2773 2787 2801 2815 2829 2843 2857 2871 2885 2899 2913 2927 /      *
*                  *   91.    igrped=3 resdmt=0                                                          *
*                  *   92.    rsfe=4r1.0;                                                                *
*                  *   93.   t                                                                           *
*                  *                                                                                     *
*                  ***************************************************************************************
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                                    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
*
*                        ***************************************************************************
*                        *                                                                         *
*                        *  3D model of a small FBR.                                               *
*                        * Cross sections from input...                                            *
*                        *                                                                         *
*                        ***************************************************************************
************************
*key end   block i read*
************************
*****************************************************************************************************************************************

*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                              ...block i - controls and dimensions...
*
*****************************************************************************************************************************************
*
*
*                                                 ...dimensions (array name = dimens)...
*
*                       14  igeom   14/15 x-y-z/ r-z-theta
*                        4  ngroup  number of energy groups
*                        8  isn     angular quadrature order
*                        5  niso    number of input isotopes (from isotxs, grupxs, or cards)
*                        5  mt      number of permanent materials
*                        5  nzone   number of zones
*                        8  im      number of coarse mesh x intervals
*                       14  it      number of fine mesh intervals
*                        8  jm      number of coarse mesh y intervals
*                       14  jt      number of fine mesh y intervals
*                        4  km      number of coarse mesh z intervals
*                       30  kt      number of fine mesh z intervals
*
*                                                    ...storage...
*
*                                                 maxlcm=  600000
*                                                 maxscm=  300000
*
*****************************************************************************************************************************************
******************************
*key end   block ii read-geom*
******************************
*
*                      14  igeom    1/2/3/6/7/8/9/11/14
******************************
*key end   block iii read-xs *
******************************
```

```
...block iii - cross section library...

...library source...
lib=cdnimp

...card library parameters (array name = cards)...

0 maxord  maximum legendre order to be found in input cross sections
8 ihm     last position in cross section table
4 iht     position of total cross section
5 ihs     position of self scatter cross section
2 ifido   -1/0/1/2 - dtf(4e18.0)/dtf/fixed fido/free fido library
1 ititl   0/1 - no/yes there is a title card before each table
0 i2lp1   0/1 - no/yes library higher order scattering contains 2l+1 factor
0 savbxs  0/1 - no/yes save binary xslib file (filename=bxslib)
0 kwikcrd 0/1 - full fido read/quick fido read (default=quick)

...energy structure...

| group | chi | vel | lower bound | upper bound | group | chi | vel | lower bound | upper bound |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 3 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 |
| 2 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 4 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 |

last neutron group(lng) is number 4

0 balxs -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance

...edit position names...

| position | edname | card position |
|---|---|---|
| 1 | chi | |
| 2 | nusigf | 3 |
| 3 | total | 4 |
| 4 | abs | 2 |
| 5 | edit1 | 1 |

*key start card libe read *

...header cards from the card library...

| isotope number | isotope name | order | header card |
|---|---|---|---|
| 1. | iso1 | p0 | - core |
| 2. | iso2 | p0 | - axblkt |
| 3. | iso3 | p0 | - nacrod |
| 4. | iso4 | p0 | - cntrod |
| 5. | iso5 | p0 | - radblk |

*key end card libe read *

*key end block iv read-mats*
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                        ... mixing instructions ...
*****************************************************************************************************************************************
*
*          mix      comp       density    comp       density etc.
*          ---      ----       -------    ----       -------
*
*     matls
*     -----
*
*       1. iso1     iso1     1.00000E+00,
*       2. iso2     iso2     1.00000E+00,
*       3. iso3     iso3     1.00000E+00,
*       4. iso4     iso4     1.00000E+00,
*       5. iso5     iso5     1.00000E+00,
************************
*key start mix card xs *
************************
************************
*key end   mix card xs *
************************
*****************************************************************************************************************************************
******************************
*key end   block v read-solvr*
******************************
******************************
*key end   block vi read-edit*
******************************
************************
*key end   input module*
************************
*
```

```
*****************************************************************************************************************************************
*****************************************************************************************************************************************
                    this threedant problem run on 01/17/95 with solver version 10-13-94beta---- release 2.5       machine e
* 3D model of a small FBR.
*Cross sections from input...
*
*****************************************************************************************************************************************
*
*                                        ...block v — solver input...
*****************************************************************************************************************************************
*          raw     as
*         input  defaulted
*
*                                        ...required input(array name = solin)...
*
*          1         1       ievt    0/1/2/3/4 - type of calculation
*                                       0 inhomogeneous source
*                                       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)
*          1         1       ibl     0/1/3 - left boundary condition
*                                       vacuum/reflective/white
*          0         0       ibr     0/1/3 - right boundary condition
*                                       vacuum/reflective/white
*          0         0       ibt     0/1/2/3 - top boundary condition
*                                       vacuum/reflective/periodic/white
*          1         1       ibb     0/1/2/3 - bottom boundary condition
*                                       vacuum/reflective/periodic/white
*          0         0       ibbk    0/1/2/3 - back boundary condition
*                                       vacuum/reflective/periodic/white
*          0         0       ibfr    0/1/2/3 - front boundary condition
*                                       vacuum/reflective/periodic/white
*
*
*                                        ...convergence controls(array name = iter)...
*
*      5.000E-05 5.000E-05   epsi    inner iteration convergence criterion (default=0.0001)
*          1         1       iitl    maximum number of inner iterations per group until fission source is near
*                                    convergence, i.e. lambda is near convergence. (default=1)
*          0        30       iitm    maximum number of inner iterations per group when close to fission source convergence
*                                    (default calculated)
*         11        11       oitm    maximum number of outer iterations (default=20)
*          0         0       itlim   iteration time limit (seconds)
*
*
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*                                    ...block v — solver input (continued)...
*
*****************************************************************************************************************************************
*          raw     as
*         input  defaulted
*
*
*                                        ...miscellaneous parameters(array name = misc)...
*
*      1.000E+00 1.000E+00   norm    normalization factor
*
*          0         0       nosigf set fissions zero when source calculation (0/1 no/yes)
*          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       nodal  0/1/2 no/cl/11 nodal option for xyz
*          0         1       iquad  -3/-2/1/2/3 - source of quadrature constants (default=1)
*                                    -3 sn2on file
*                                    -2 hybrid product set (triangular arrangement)
*                                     1 old twotran built-in set
*                                     2 product set (rectangular arrangement)
*                                     3 card input
*
*
*                                           ...output controls(array name = solout)...
*
*                        0  fluxp  0/1/2   none/isotropic/all moments - flux print
*                        2  xsectp 0/1/2   none/principal/all - macroscopic cross section print
*                        0  fissrp 0/1     no/yes - print final fission source rate
*                        0  sourcp 0/1/2/3 no/as read/normalized/both - print inhomogeneous source
*                        0  angp   0/1     no/yes - print angular fluxes
*                        0  raflux 0/1     no/yes - write angular fluxes to file raflxm or aaflxm(if ith=1)
*                        0  rmflux 0/1     no/yes - write flux moments to file rmflux
*                        0  balp   0/1     no/yes - print coarse mesh balances
*
*
*                                           ...parameters inferred from input arrays...
*
*                                  1  inchi  0/1/2 none/one chi/zonewise chi
*                                  0  isdenx 0/1/n - none/x density vector/full matrix
*                                  0  isdeny 0/1 no/yes - use y density vector
*                                  0  iqan   source anisotropy
*                                  0  isorse number of source moments input
*                                  0  isorsx number of sourcx moments input
*                                  0  isorsy number of sourcy moments input
*                                  0  isorsz number of sourcz moments input
*                                  0  isorsf number of sourcf moments input
*                                  0  iql    -1/0/1/2 isotropic/none/all angles/vectors -left boundary source
*                                  0  iqr    -1/0/1/2 isotropic/none/all angles/vectors -right boundary source
*                                  0  iqt    -1/0/1/2 isotropic/none/all angles/vectors -top boundary source
*                                  0  iqb    -1/0/1/2 isotropic/none/all angles/vectors -bottom boundary source
*                                  0  iqf    -1/0/1/2 isotropic/none/all angles/vectors -front boundary source
*                                  0  iqk    -1/0/1/2 isotropic/none/all angles/vectors -back boundary source
```

```
*
*
*
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*
*                                   ...parameters from block i...
*
*                    14  igeom   14/15  x-y-z/r-z-theta
*                     4  ngroup  number of energy groups
*                     8  isn     angular quadrature order
*                     5  mt      number of permanent materials
*                     5  nzone   number of zones
*                     8  im      number of coarse mesh x intervals
*                     8  jm      number of coarse mesh y intervals
*                     4  km      number of coarse mesh z intervals
*                    14  it      number of fine mesh x intervals
*                    14  jt      number of fine mesh y intervals
*                    30  kt      number of fine mesh z intervals
*
*
*
*
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*
*                                        ...material assignments to zones...
*
*
*******************************
*key start matls to zones     *
*******************************
*
*                                                zone     material
*                                              no. name   no. name
*
*                                               1 zone1    1 iso1
*                                               2 zone2    2 iso2
*                                               3 zone3    3 iso3
*                                               4 zone4    4 iso4
*                                               5 zone5    5 iso5
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
*
************************
*key start storage map *
************************
*
*  scm storage summary...
*
*     total scm required for this problem       90719
*     maximum scm available (maxscm= )         300000
*     scm required for transport                67970
*     scm required for diffusion                90682
*
*
*  lcm storage summary...
*
*     lcm selected for this problem                  203602
*     maximum lcm specified (maxlcm= )               600000
*          storage required for all quantities  in core is      203602
*
*          diffusion params are in core.  if o.k. to put in core, set maxlxm .ge.       203602
*          diffusion fluxes are in core.  if o.k. to put in core, set maxlxm .ge.       115087
*          scalar tr fluxes are in core.  if o.k. to put in core, set maxlxm .ge.        94162
*          storage required for all quantities  on disk is       73162
*


      203602 words lcm required lcmadd
**********************************************************************************************************************************
**********************************************************************************************************************************
*
* zero flux
*
**********************************************************************************************************************************
*
************************
*key start sn constants*
************************
*
* s 8 constants for groups  1 to  4
*                 mu              eta             xi              weight
*
*     1      0.19232747E+00  0.96229948E+00  0.19232747E+00  0.14598559E-01
*     2      0.57735027E+00  0.79352178E+00  0.19232746E+00  0.11656904E-01
*     3      0.19232747E+00  0.79352178E+00  0.57735027E+00  0.11656904E-01
*     4      0.79352178E+00  0.57735027E+00  0.19232746E+00  0.11656904E-01
*     5      0.57735027E+00  0.57735027E+00  0.57735027E+00  0.11262900E-01
*     6      0.19232747E+00  0.57735027E+00  0.79352178E+00  0.11656904E-01
*     7      0.96229948E+00  0.19232747E+00  0.19232747E+00  0.14598559E-01
*     8      0.79352178E+00  0.19232747E+00  0.57735027E+00  0.11656904E-01
*     9      0.57735027E+00  0.19232747E+00  0.79352178E+00  0.11656904E-01
*    10      0.19232747E+00  0.19232747E+00  0.96229948E+00  0.14598559E-01
```

```
*************************************************************material map*************************************************************
*************************
*kay start material map*
*************************
* material map for k mesh intervals   1, to   4 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
*
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 2 2 2 5 5 5 5 5 5 5 5 5 5 5
* 2 2 2 2 2 2 5 5 5 5 5 5 5 5
* 2 2 2 2 2 2 2 2 5 5 5 5 5 5
* 2 2 2 2 2 2 2 2 2 5 5 5 5 5
* 2 2 2 2 2 2 2 2 2 5 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 2 5 5 5
* 2 2 2 2 2 2 2 2 2 2 2 5 5 5
* 2 2 2 2 2 2 2 3 3 2 2 5 5 5

* material map for k mesh intervals   5, to  15 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
*
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 1 1 1 5 5 5 5 5 5 5 5 5 5 5
* 1 1 1 1 1 1 5 5 5 5 5 5 5 5
* 1 1 1 1 1 1 1 1 5 5 5 5 5 5
* 1 1 1 1 1 1 1 1 1 5 5 5 5 5
* 1 1 1 1 1 1 1 1 1 5 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 1 5 5 5
* 1 1 1 1 1 1 1 1 1 1 1 5 5 5
* 1 1 1 1 1 1 1 3 3 1 1 5 5 5
* material map for k mesh intervals  16, to  26 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
*
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 1 1 1 5 5 5 5 5 5 5 5 5 5 5
* 1 1 1 1 1 1 5 5 5 5 5 5 5 5
* 1 1 1 1 1 1 1 1 5 5 5 5 5 5
* 1 1 1 1 1 1 1 1 1 5 5 5 5 5
* 1 1 1 1 1 1 1 1 1 5 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 5 5 5 5
* 1 1 1 1 1 1 1 1 1 1 1 5 5 5
* 1 1 1 1 1 1 1 1 1 1 1 5 5 5
* 1 1 1 1 1 1 1 3 3 1 1 5 5 5
* material map for k mesh intervals  27, to  30 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
*
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 5 5 5 5 5 5 5 5 5 5 5 5 5 5
* 2 2 2 5 5 5 5 5 5 5 5 5 5 5
* 2 2 2 2 2 2 5 5 5 5 5 5 5 5
* 2 2 2 2 2 2 2 2 5 5 5 5 5 5
* 2 2 2 2 2 2 2 2 2 5 5 5 5 5
* 2 2 2 2 2 2 2 2 2 5 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 5 5 5 5
* 2 2 2 2 2 2 2 2 2 2 2 5 5 5
* 2 2 2 2 2 2 2 2 2 2 2 5 5 5
* 2 2 2 2 2 2 2 3 3 2 2 5 5 5
*
**************************************************************************************************************************************
**************************************************************************************************************************************
```

```
*****************************************************************************************************************************
*
*                                        Three-dimensional coarse mesh geometry edit...
*
*****************************************************************************************************************************
*
*
*****************************
*key start geometry edit    *
*****************************
*   n    xmesh       xint   deltax       ymesh       yint   deltay       zmesh       zint   deltaz
*
*   0  0.000E+00                       0.000E+00                       0.000E+00
*   1
       1.500E+01       3   5.000E+00
                                       5.000E+00       1   5.000E+00
                                                                       2.000E+01       4   5.000E+00
*   2
       3.000E+01       3   5.000E+00
                                       1.500E+01       2   5.000E+00
                                                                       7.500E+01      11   5.000E+00
*   3
       3.500E+01       1   5.000E+00
                                       3.000E+01       3   5.000E+00
                                                                       1.300E+02      11   5.000E+00
*   4
       4.000E+01       1   5.000E+00
                                       4.000E+01       2   5.000E+00
                                                                       1.500E+02       4   5.000E+00
*   5
       4.500E+01       1   5.000E+00
                                       4.500E+01       1   5.000E+00
*   6
       5.000E+01       1   5.000E+00
                                       5.000E+01       1   5.000E+00
*   7
       5.500E+01       1   5.000E+00
                                       5.500E+01       1   5.000E+00
*   8
       7.000E+01       3   5.000E+00
                                       7.000E+01       3   5.000E+00
*
*****************************************************************************************************************************
```

```
...cross section related data from file macros  01/17/09:46: version  1 ...

 1 iso1      2 iso2      3 iso3      4 iso4      5 iso5

...cross sections for legendre orders up to p0...

*key start mac cross sections*

group  1

...principal cross sections...

                zone          chi          nu*fission      total        absorption
              no.  name
               1   zone1     5.8332E-01    2.0606E-02    1.1457E-01    7.4555E-03
               2   zone2     5.8332E-01    1.3177E-02    1.1649E-01    5.3542E-03
               3   zone3     5.8332E-01    0.0000E+00    6.5898E-02    3.1074E-04
               4   zone4     5.8332E-01    0.0000E+00    1.8433E-01    5.9764E-03
               5   zone5     5.8332E-01    1.8950E-02    1.1965E-01    7.4328E-03

                                  ...scattering matrices...
                                    (21+1 not included)

  zone   order  first grp    cross sections
    1      0      1          7.0433E-02
    2      0      1          7.1604E-02
    3      0      1          4.7441E-02
    4      0      1          1.3437E-01
    5      0      1          6.9116E-02

group  2

...principal cross sections...

                zone          chi          nu*fission      total        absorption
              no.  name
               1   zone1     4.0545E-01    6.1057E-03    2.0518E-01    3.5254E-03
               2   zone2     4.0545E-01    1.2603E-04    2.2052E-01    1.4860E-03
               3   zone3     4.0545E-01    0.0000E+00    1.0981E-01    1.1306E-04
               4   zone4     4.0545E-01    0.0000E+00    3.6612E-01    1.7694E-02
               5   zone5     4.0545E-01    1.7526E-04    2.4219E-01    1.9906E-03

                                  ...scattering matrices...
                                    (21+1 not included)

  zone   order  first grp    cross sections
    1      0      2          1.9544E-01    3.4797E-02
    2      0      2          2.1044E-01    3.7317E-02
    3      0      2          1.0614E-01    1.7689E-02
    4      0      2          3.1858E-01    4.3778E-02
    5      0      2          2.3063E-01    4.0413E-02

group  3

...principal cross sections...

                zone          chi          nu*fission      total        absorption
              no.  name
               1   zone1     1.1231E-02    6.9140E-03    3.2938E-01    7.8014E-03
               2   zone2     1.1231E-02    1.5238E-04    3.4454E-01    5.3530E-03
               3   zone3     1.1231E-02    0.0000E+00    1.8676E-01    4.4899E-04
               4   zone4     1.1231E-02    0.0000E+00    6.1553E-01    8.8274E-02
               5   zone5     1.1231E-02    2.0698E-04    3.5648E-01    6.7904E-03

                                  ...scattering matrices...
                                    (21+1 not included)

  zone   order  first grp    cross sections
    1      0      3          3.2059E-01    6.2086E-03    1.8828E-03
    2      0      3          3.3751E-01    8.5985E-03    2.2171E-03
    3      0      3          1.8530E-01    3.5547E-03    4.5701E-04
    4      0      3          5.1959E-01    2.9843E-02    2.0605E-04
    5      0      3          3.4841E-01    9.5703E-03    2.6862E-03

group  4

...principal cross sections...

                zone          chi          nu*fission      total        absorption
              no.  name
               1   zone1     0.0000E+00    2.6069E-02    3.8981E-01    2.7450E-02
               2   zone2     0.0000E+00    7.8730E-04    3.8836E-01    1.3469E-02
               3   zone3     0.0000E+00    0.0000E+00    2.0993E-01    1.0752E-03
               4   zone4     0.0000E+00    0.0000E+00    1.0949E+00    4.7659E-01
               5   zone5     0.0000E+00    1.1345E-03    3.7943E-01    1.5802E-02

                                  ...scattering matrices...
                                    (21+1 not included)

  zone   order  first grp    cross sections
    1      0      4          3.6236E-01    9.9297E-04    7.0721E-07
```

```
*    2      0      4          3.7489E-01    1.6853E-03    6.6830E-07
*    3      0      4          2.0886E-01    1.0128E-03    1.7760E-07
*    4      0      4          6.1827E-01    7.6621E-03    8.7119E-07
*    5      0      4          3.6363E-01    1.2720E-03    1.9957E-07
*
*
*************************************************************************************************************************************
*************************************************************************************************************************************
```

```
*************************************************************************************************************************
*************************************************************************************************************************
*
*                                        ...iteration controls and criteria...
*
*************************************************************************************************************************
*
*
*                                          ***iteration criteria***
*
*                                                transport inners
*
*          criterion    quantity to test                                  value     action taken if value exceeded
*          ---------    ----------------                                  -----     ------------------------------
*             iitl  - inner iteration count until near lambda              1        terminates inners
*                       (i.e. fission source) convergence
*             iitm  - inner iteration count when near lambda              30        terminates inners
*                       (i.e. fission source) convergence
*             epsi  - fractional ptwise flux change                    5.00E-05    does another inner
*                       per inner
*
*                                             diffusion sub-outers
*
*          criterion    quantity to test                                  value     action taken if value exceeded
*          ---------    ----------------                                  -----     ------------------------------
*             oitmd - sub-outer iteration count                           22       terminates sub-outers
*             eps   - diffusion lambda-1.0 (see note below)            5.00E-05    does another sub-outer
*             eps   - fractional ptwise fission change                 5.00E-05    does another sub-outer
*                       per sub-outer (see note below)
*
*             note: eps, when the problem is finally converged, will equal epsi, the value shown above. however,
*                   early in the iteration process, a larger value may be used to avoid unnecessary iterations.
*
*                                           final convergence criteria
*
*          criterion    quantity to test                                  value     action taken if value exceeded
*          ---------    ----------------                                  -----     ------------------------------
*             oitm  - outer iteration count                               11       quits with error message
*             epsi  - transport lambda-1.0                             5.00E-05    does another outer
*
*************************************************************************************************************************
*************************************************************************************************************************

*************************************************************************************************************************
*************************************************************************************************************************
*
*                        ...flux and eigenvalue convergence as monitored by threedant...
*
*************************************************************************************************************************
*****************************
*key start iteration monitor *
*****************************
*
*
* cpu time outer       diffusion      k-eff                      max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  eigenvalue    lambda-1        flux change  fiss change converged
*   4.41     0
*   5.76     1      0       5      0.94651641  8.53036E-03   0.00000E+00  0.00000E+00  **no**
*   7.54     2      4       3      0.97258730  2.03948E-02   2.44013E-01  2.59746E-01  **no**
*   9.27     3      4       3      0.97336421  3.52853E-05   3.43649E-02  3.19468E-02  **no**
*  10.76     4      4       2      0.97344585 -8.68557E-06   6.07123E-03  5.14875E-03  **no**
*  12.25     5      4       2      0.97347159  7.40002E-06   2.54048E-03  9.46899E-04  **no**
*  13.74     6      4       2      0.97347769  1.83401E-06   1.10209E-03  2.30904E-04  **no**
*
*    ---------------------------------------------------------------------------
*
*              — inner iteration summary for outer iteration no.   7 —
*
*                                  iter per max flux   at
*                                group  group   change   mesh
*                                  1      2   0.80E-05  13, 12,  1
*                                  2      2   0.14E-04  13,  1,  3
*                                  3      1   0.46E-04   9,  3, 30
*                                  4      4   0.45E-04   9,  3, 25
*
*
* cpu time outer       diffusion      k-eff                      max ptwise   max ptwise   inners
*  (sec)    no. inners sub-outers  eigenvalue    lambda-1        flux change  fiss change converged
*  16.52     7      9       2      0.97347913  2.96091E-07   4.59971E-05  5.09646E-06   yes
*
*                $$$$$$ all convergence criteria satisfied $$$$$$
*
*  particle balance =  3.81337E-08      total inners all outers =   29
*
*************************************************************************************************************************
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                        ...group edit and balances upon convergence...
*
**********************************************************************************************************************************
*
*
*                        ...title--- 3D model of a small FBR.                                        ...
*
*
*                                        ...system balance tables... (neutrons only)
*
*
*******************************
*key start balance table     *
*******************************
*
*
*  gp          source     fission source        absorption        in scatter      self scatter      out scatter       net leakage
*
*   1   0.0000000E+00     5.8331900E-01     9.3606434E-02    -2.5934810E-13     9.0152168E-01     4.7787527E-01     1.1837289E-02
*   2   0.0000000E+00     4.0545000E-01     2.4701759E-01     4.5277070E-01     1.5724966E+01     5.3335904E-01     7.7843956E-02
*   3   0.0000000E+00     1.1231000E-02     4.3338687E-01     5.5836861E-01     1.9304294E+01     6.5330805E-02     7.0881773E-02
*   4   0.0000000E+00     0.0000000E+00     6.0224756E-02     6.5357654E-02     9.8799113E-01     1.2076664E-06     5.1316767E-03
* *
* tot   0.0000000E+00     1.0000000E+00     8.3423565E-01     1.0764970E+00     3.6918773E+01     1.0765663E+00     1.6569469E-01
*
*
*
*  gp   right leakage horizontl leakage       top leakage  vertical leakage      back leakage   fr-back leakage  particle balance
*
*   1   3.9859736E-03     3.9859736E-03     4.1657121E-03     4.1657121E-03     1.8428040E-03     3.6856031E-03     3.9000696E-09
*   2   2.5613073E-02     2.5613073E-02     2.6512387E-02     2.6512387E-02     1.2859263E-02     2.5718496E-02     5.4039212E-08
*   3   2.3914620E-02     2.3914620E-02     2.4460558E-02     2.4460558E-02     1.1253313E-02     2.2506595E-02     5.3114597E-08
*   4   1.5826589E-03     1.5826589E-03     1.6026469E-03     1.6026469E-03     9.7318647E-04     1.9463709E-03     4.2508290E-09
* *
* tot   5.5096326E-02     5.5096326E-02     5.6741304E-02     5.6741304E-02     2.6928566E-02     5.3857065E-02     3.8133656E-08
*
*
*
*  gp    left leakage    bottom leakage     front leakage     nprod spectrum
*
*   1   1.6653345E-16    -4.7184479E-16     1.8427991E-03     2.6283041E-01
*   2   9.9920072E-16    -2.7755576E-15     1.2859233E-02     3.9058651E-01
*   3   7.7715612E-16    -2.4424907E-15     1.1253282E-02     3.0268953E-01
*   4   3.4694470E-17    -1.3877788E-16     9.7318439E-04     4.3893544E-02
*
*
*tot    1.9775848E-15    -5.8286709E-15     2.6928498E-02     1.0000000E+00
*
**********************************************************************************************************************************

**********************************************************************************************************************************
**********************************************************************************************************************************
*
*     Multigrid work units...  Total=    959.20 WU.
*          By group...
*     1    242.27  2    245.49  3    236.05  4    235.39
*
*     Multigrid average convergence rate by group...
*     1 0.5065  2 0.5319  3 0.5087  4 0.5358
*
*     timing info...tswep,tdsa,trelx,tput3,tintrp=    5.00    6.84    3.58    1.03    0.73 seconds.

integral summary information
summary integral-k-eff     9.7347913E-01
integral-source-i              neutron  0.0000000E+00
integral-fission-i             neutron  1.0000000E+00
integral-absorption-i          neutron  8.3423565E-01
integral-in-scak-i             neutron  1.0764970E+00
integral-self-scak-i           neutron  3.6918773E+01
integral-out-scak-i            neutron  1.0765663E+00
integral-net lkage-i           neutron  1.6569469E-01
integral-right lkage-i         neutron  5.5096326E-02
integral-horizontal lkage-i    neutron  5.5096326E-02
integral-top lkage-i           neutron  5.6741304E-02
integral-vertical lkage-i      neutron  5.6741304E-02
integral-back lkage-i          neutron  2.6928566E-02
integral-fr-back lkage-i       neutron  5.3857065E-02
integral-particle bal-i        neutron  1.1530471E-07
*
*
*                                        ...interface file rtflux written..

*
*
*                                        ...interface file snconn written..


threedant iteration time, mins  2.7624E-01
```

```
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                  edit run on 01/17/95 with solver version 10-13-94*product release 2.5      machine e
*
**********************************************************************************************************************************
**********************************************************************************************************************************
**********************************************************************************************************************************
*
*                                                ...edit output...
*
**********************************************************************************************************************************
*
*
*                                             ...block vi - edit specification data...
*
*
************************
*key start edit output *
************************


**************************************************************************************************************
*
*
* cross section balancing (balxs.ne.0)
*       or
* transport correction (trcor=diag, cesaro, or bhs)
* will NOT be reflected in edits
*
*
**************************************************************************************************************


*                                          ...input control integers...
*
*                  1  pted      0/1     no/yes - point edits desired
*                  1  zned      0/1     no/yes - zone edits desired
*
*                  0  ajed      0/1     direct/adjoint edit (use rtflux/atflux file)
*
*                  3  igrped    0/1/2/3 print totals only/print broad groups only/same as 1/print all groups and totals
*                  0  byvolp    0/1     no/yes - multiply point reaction rates by mesh volumes
*                  0  rzflux    0/1     no/yes - write the rzflux file (zone average flux file)
*                  0  rzmflx    0/1     no/yes - write the rzmflx file (zone average flux moments file)
*
*
*                                          ...floating parameters...
*
*          0.000000E+00  power     0/p     no/normalize all results, including flux files, to p megawatts
*          2.100000E+02  mevper    mev per fission (default: 210 mev)
*
*
*
*                                       ...energy related edit information...
*
*                                         4  number of fine neutron groups
*                                         0  number of fine gamma groups
*                                         4  total number of fine groups
*                                         4  total number of broad groups
*
*
*
*                                       ...space related edit information...
*
*                                        27  number of points to edit
*                                         5  number of zones to edit
*                                         0  0/1  no/yes  density factors were input
*                                         1  0/1/2/3  no/solver/edit/both  zones mass edit
*                                                (requires atomic weights to be present)
*
**********************************************************************************************************************************
**********************************************************************************************************************************


****point translation from 1d to 3d

 point     i     j     k     avg. radii      avg. y          avg. z          volume
  2745     1     1    15   2.500000E+00   2.500000E+00   7.250000E+01   1.250000E+02
  2746     2     1    15   7.500000E+00   2.500000E+00   7.250000E+01   1.250000E+02
  2747     3     1    15   1.250000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2748     4     1    15   1.750000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2749     5     1    15   2.250000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2750     6     1    15   2.750000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2751     7     1    15   3.250000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2752     8     1    15   3.750000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2753     9     1    15   4.250000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2754    10     1    15   4.750000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2755    11     1    15   5.250000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2756    12     1    15   5.750000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2757    13     1    15   6.250000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2758    14     1    15   6.750000E+01   2.500000E+00   7.250000E+01   1.250000E+02
  2759     1     2    15   2.500000E+00   7.500000E+00   7.250000E+01   1.250000E+02
  2773     1     3    15   2.500000E+00   1.250000E+01   7.250000E+01   1.250000E+02
  2787     1     4    15   2.500000E+00   1.750000E+01   7.250000E+01   1.250000E+02
  2801     1     5    15   2.500000E+00   2.250000E+01   7.250000E+01   1.250000E+02
  2815     1     6    15   2.500000E+00   2.750000E+01   7.250000E+01   1.250000E+02
  2829     1     7    15   2.500000E+00   3.250000E+01   7.250000E+01   1.250000E+02
  2843     1     8    15   2.500000E+00   3.750000E+01   7.250000E+01   1.250000E+02
  2857     1     9    15   2.500000E+00   4.250000E+01   7.250000E+01   1.250000E+02
  2871     1    10    15   2.500000E+00   4.750000E+01   7.250000E+01   1.250000E+02
  2885     1    11    15   2.500000E+00   5.250000E+01   7.250000E+01   1.250000E+02
  2899     1    12    15   2.500000E+00   5.750000E+01   7.250000E+01   1.250000E+02
  2913     1    13    15   2.500000E+00   6.250000E+01   7.250000E+01   1.250000E+02
  2927     1    14    15   2.500000E+00   6.750000E+01   7.250000E+01   1.250000E+02
```

```
********************************************************************************
*point/zone edit information on file-edtout
********************************************************************************
```

```
*******************************
*key start response functions*
*******************************

                                 ******************************
                                 *                            *
                                 *       reaction rates       *
                                 *            from            *
                                 *     response functions     *
                                 *                            *
                                 ******************************


                                ...point edit for broad group   1 ...


                                    ... response functions ...

                                 point     av rad       rspf1

                                  2745     2.5000   9.20579E-05
                                  2746     7.5000   9.05611E-05
                                  2747    12.5000   8.75670E-05
                                  2748    17.5000   8.31213E-05
                                  2749    22.5000   7.71110E-05
                                  2750    27.5000   6.97132E-05
                                  2751    32.5000   5.85422E-05
                                  2752    37.5000   4.55062E-05
                                  2753    42.5000   3.88194E-05
                                  2754    47.5000   3.53697E-05
                                  2755    52.5000   2.60786E-05
                                  2756    57.5000   1.39354E-05
                                  2757    62.5000   6.71212E-06
                                  2758    67.5000   3.47125E-06
                                  2759     2.5000   9.07554E-05
                                  2773     2.5000   8.82229E-05
                                  2787     2.5000   8.44229E-05
                                  2801     2.5000   7.95213E-05
                                  2815     2.5000   7.33623E-05
                                  2829     2.5000   6.62376E-05
                                  2843     2.5000   5.82539E-05
                                  2857     2.5000   4.92469E-05
                                  2871     2.5000   3.96686E-05
                                  2885     2.5000   2.78370E-05
                                  2899     2.5000   1.49694E-05
                                  2913     2.5000   7.15064E-06
                                  2927     2.5000   3.71338E-06


                                ...zone edit for broad group   1 ...


                                    ... response functions ...

                                 zone     volume       rspf1

                                    1 2.6400E+05   1.10036E+01
                                    2 9.6000E+04   4.86814E-01
                                    3 3.7500E+03   9.32824E-02
                                    4 3.7500E+03   9.32825E-02
                                    5 3.6750E+05   1.19799E+00
                                  ------------   -----------
                                  sum 7.3500E+05   1.28750E+01


                                ...point edit for broad group   2 ...


                                    ... response functions ...

                                 point     av rad       rspf1

                                  2745     2.5000   5.06739E-04
                                  2746     7.5000   4.98458E-04
                                  2747    12.5000   4.81972E-04
                                  2748    17.5000   4.57496E-04
                                  2749    22.5000   4.25019E-04
                                  2750    27.5000   3.84812E-04
                                  2751    32.5000   3.33990E-04
                                  2752    37.5000   2.85404E-04
                                  2753    42.5000   2.50305E-04
                                  2754    47.5000   2.11018E-04
                                  2755    52.5000   1.61176E-04
                                  2756    57.5000   1.08919E-04
                                  2757    62.5000   6.49290E-05
                                  2758    67.5000   3.01813E-05
                                  2759     2.5000   4.99656E-04
                                  2773     2.5000   4.85592E-04
                                  2787     2.5000   4.64718E-04
                                  2801     2.5000   4.37329E-04
                                  2815     2.5000   4.03825E-04
                                  2829     2.5000   3.64799E-04
                                  2843     2.5000   3.21000E-04
                                  2857     2.5000   2.73186E-04
                                  2871     2.5000   2.22573E-04
                                  2885     2.5000   1.69708E-04
                                  2899     2.5000   1.14813E-04
                                  2913     2.5000   6.83760E-05
                                  2927     2.5000   3.18106E-05


                                ...zone edit for broad group   2 ...


                                    ... response functions ...
```

zone	volume	rspf1
1	2.6400E+05	6.18625E+01
2	9.6000E+04	4.42842E+00
3	3.7500E+03	6.09456E-01
4	3.7500E+03	6.09456E-01
5	3.6750E+05	1.11571E+01
sum	7.3500E+05	7.86669E+01

...point edit for broad group 3 ...

... response functions ...

point	av rad	rspf1
2745	2.5000	3.24537E-04
2746	7.5000	3.19436E-04
2747	12.5000	3.09389E-04
2748	17.5000	2.94598E-04
2749	22.5000	2.75604E-04
2750	27.5000	2.52855E-04
2751	32.5000	2.27665E-04
2752	37.5000	2.05602E-04
2753	42.5000	1.84614E-04
2754	47.5000	1.55344E-04
2755	52.5000	1.24648E-04
2756	57.5000	9.60691E-05
2757	62.5000	6.47537E-05
2758	67.5000	2.99722E-05
2759	2.5000	3.20058E-04
2773	2.5000	3.11175E-04
2787	2.5000	2.98044E-04
2801	2.5000	2.80924E-04
2815	2.5000	2.60164E-04
2829	2.5000	2.36326E-04
2843	2.5000	2.09953E-04
2857	2.5000	1.82077E-04
2871	2.5000	1.53428E-04
2885	2.5000	1.26294E-04
2899	2.5000	9.83776E-05
2913	2.5000	6.67183E-05
2927	2.5000	3.09541E-05

...zone edit for broad group 3 ...

... response functions ...

zone	volume	rspf1
1	2.6400E+05	4.21663E+01
2	9.6000E+04	4.41979E+00
3	3.7500E+03	4.62737E-01
4	3.7500E+03	4.62737E-01
5	3.6750E+05	1.18341E+01
sum	7.3500E+05	5.93457E+01

...point edit for broad group 4 ...

... response functions ...

point	av rad	rspf1
2745	2.5000	1.11768E-05
2746	7.5000	1.10052E-05
2747	12.5000	1.06823E-05
2748	17.5000	1.02056E-05
2749	22.5000	9.67723E-06
2750	27.5000	9.14910E-06
2751	32.5000	9.31915E-06
2752	37.5000	1.00367E-05
2753	42.5000	9.36354E-06
2754	47.5000	7.10948E-06
2755	52.5000	5.55603E-06
2756	57.5000	4.92485E-06
2757	62.5000	3.73025E-06
2758	67.5000	1.80281E-06
2759	2.5000	1.10228E-05
2773	2.5000	1.07192E-05
2787	2.5000	1.02707E-05
2801	2.5000	9.69048E-06
2815	2.5000	8.98989E-06
2829	2.5000	8.20081E-06
2843	2.5000	7.34151E-06
2857	2.5000	6.48765E-06
2871	2.5000	5.68563E-06
2885	2.5000	5.21664E-06
2899	2.5000	4.84458E-06
2913	2.5000	3.75220E-06
2927	2.5000	1.82019E-06

...zone edit for broad group 4 ...

... response functions ...

zone	volume	rspf1
1	2.6400E+05	1.59649E+00
2	9.6000E+04	3.50324E-01
3	3.7500E+03	2.52511E-02
4	3.7500E+03	2.52511E-02
5	3.6750E+05	7.35933E-01
sum	7.3500E+05	2.73325E+00

...point edit for the sum of the neutron groups ...

... response functions ...

point	av rad	rspf1
2745	2.5000	9.34510E-04
2746	7.5000	9.19460E-04
2747	12.5000	8.89609E-04
2748	17.5000	8.45421E-04
2749	22.5000	7.87412E-04
2750	27.5000	7.16529E-04
2751	32.5000	6.29516E-04
2752	37.5000	5.46548E-04
2753	42.5000	4.83101E-04
2754	47.5000	4.08840E-04
2755	52.5000	3.17459E-04
2756	57.5000	2.23848E-04
2757	62.5000	1.40125E-04
2758	67.5000	6.54276E-05
2759	2.5000	9.21492E-04
2773	2.5000	8.95710E-04
2787	2.5000	8.57456E-04
2801	2.5000	8.07464E-04
2815	2.5000	7.46342E-04
2829	2.5000	6.75563E-04
2843	2.5000	5.96548E-04
2857	2.5000	5.10998E-04
2871	2.5000	4.21355E-04
2885	2.5000	3.29056E-04
2899	2.5000	2.33005E-04
2913	2.5000	1.45997E-04
2927	2.5000	6.82982E-05

...zone edit for the sum of the neutron groups ...

... response functions ...

zone	volume	rspf1
1	2.6400E+05	1.16629E+02
2	9.6000E+04	9.68535E+00
3	3.7500E+03	1.19073E+00
4	3.7500E+03	1.19073E+00
5	3.6750E+05	2.49251E+01
sum	7.3500E+05	1.53621E+02

```
*key start run highlights     *
```

run highlights

```
*****************************************
*                                       *
*    all modules are tentatively go.    *
*     interface file geodst written.    *
*       cross sections from cards.      *
*    *kwikrd overridden for ifido=2*    *
*    interface mixing files written.    *
*     interface file asgmat written     *
*    xs files macxs,smxedt written.     *
*     interface file solinp written.    *
*     interface file editit written.    *
*        start solver execution.        *
*     all convergence criteria met.     *
*     interface file rtflux written.    *
*     interface file snconn written.    *
*          start edit execution.        *
*              edits completed.         *
*                                       *
*****************************************
```

storage and timing history

module	scm words	scm limit	lcm words	lcm limit	cpu seconds	sysio seconds
0	0	0	0	0	16.8	1.4
100	9548	300000	0	0	4.3	0.2
101	0	0	0	0	0.7	0.0
102	1115	300000	289	600000	0.0	0.0
103	72	300000	0	0	0.1	0.0
104	2077	300000	0	0	0.1	0.1
105	0	0	0	0	0.0	0.0
106	0	0	0	0	0.0	0.0
107	235	300000	190	600000	0.1	0.1
108	1068	300000	6	600000	0.1	0.0
109	15442	300000	0	0	0.1	0.0
112	0	0	0	0	0.0	0.0
200	90719	300000	203602	600000	12.2	1.0
201	0	0	0	0	0.1	0.1
202	0	0	0	0	0.0	0.0
203	1393	300000	0	0	0.0	0.0
204	0	0	0	0	0.0	0.0
205	0	0	0	0	0.0	0.0
206	0	0	0	0	7.5	0.2
207	0	0	0	0	4.6	0.6
208	0	0	0	0	0.0	0.0
210	0	0	0	0	0.0	0.1
211	0	0	0	0	0.0	0.0
300	0	0	0	0	0.3	0.2
301	36090	300000	133	600000	0.1	0.1
302	0	0	0	0	0.0	0.0
400	0	0	0	0	0.0	0.0

...execution terminated...

APPENDIX B: OPERATING SYSTEM SPECIFICS

UNIX/UNICOS Execution

On UNIX or UNICOS systems, the input is on STDIN and the printed output is on STDOUT. Thus, the user will normally cause execution of the program with the command:

dant.x < *odninp* > *odnout*

where - dant.x is the name of the executable file, *odninp* is the user's choice for a name for the input file, and *odnout* is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is dant.x.

STDERR contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDERR are any error messages.

Library Search Path

Most files read or written by THREEDANT are in the current UNIX working directory. Some forms of cross-section files may be kept in other directories. By setting the environment variable SNXSPATH, the user may specify an ordered set of alternate directories in which the program should look for the named files. As an example, if an ISOTXS file is in the directory, `/usr/tmp/xs`, then the following command can be used

```
setenv SNXSPATH /usr/tmp/xs
```

and THREEDANT will then look in that named directory for the library. The search path for each of the possible libraries is given in Table 6.2.

Table 6.2 UNIX Search Path

LIB	SEARCH PATH
MACRXS	Current Working Directory (CWD).
GRUPXS	**SNXSPATH**, then CWD.
ISOTXS	**SNXSPATH**, then CWD.
BXSLIB	**SNXSPATH**, then CWD, but see text below.
ODNINP	None, the library is contained in the input file.
MACBCD	CWD
XSLIBB	CWD
MENDF[a]	Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.
MENDFG[b]	
XSLIB	**SNXSPATH**, then CWD
other	For any name other than those above, the program will assume the form is XSLIB and search for it in **SNXSPATH**, then CWD.

a. Available only at Los Alamos.

b. Available only at Los Alamos.

SNXSPATH can be used to protect an input BXSLIB file from being overwritten. See the discussion on page 6-40.

DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

7

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This chapter is intended to provide a more detailed description of a portion of the input than is provided in the User's Guide chapters. In particular, the input to Blocks-I, II, and V will be discussed here. Detailed discussion of the input for Blocks-III, IV, and VI are found in other separate support chapters.

In order to condense the discussion and make it simpler to understand, much of the description is couched in terms of one-dimensional geometry. The extension to two and three dimensions is quite straightforward and is briefly mentioned.

MORE DETAILS ON BLOCK-I INPUT

The input parameters provided in Block-I are used by the code to determine storage requirements for the problem being run (the code uses variable dimensioning), to provide error checking on subsequent input, and/or to control the execution-flow of the code.

In this section are provided details on certain of the parameters that appear in the Block-I input for ONEDANT. In some cases, the "details" are simply references to other chapters of this manual.

Angular Quadrature (ISN)

The numerical value (an even integer) entered for the parameter ISN is simply the value of N in S_n, that is, the angular quadrature order desired for the current calculation. The discrete-ordinates, or S_n, approximation is described in "Discrete-Ordinates Equations in One Dimension" on page 12-27. See "Input of Quadrature Sets" on page 7-24 for details on angular quadrature sets supplied in the code

Geometry (NZONE, IM, IT)

The parameter NZONE is the number of different zones that are to be defined for the calculation. See "MORE DETAILS ON GEOMETRY INPUT" on page 7-13 for the concept and meaning of the term zone.

The number of coarse spatial mesh intervals in the x direction of the problem being solved is denoted by the parameter IM. See "MORE DETAILS ON GEOMETRY INPUT" on page 7-13 for the concept and meaning of the term coarse mesh interval. Correspondingly, the number of coarse spatial mesh intervals in the y and z directions are denoted respectively by the parameters JM and KM.

The total number of fine spatial mesh intervals (or mesh points) for the problem being solved is given by the parameters IT for the x direction, JT for the y direction, and KT for the z direction. The fine mesh interval or point is described in "Discretization of the Spatial Variable" on page 12-32. See page 7-13 for the concept and meaning of the term fine mesh interval.

MAXSCM, MAXLCM

Originally designed for the CDC-7600 computer, the code is structured 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 will be automatically used by the code if LCM (or simulated LCM) storage requirements are exceeded.

The MAXSCM parameter allows the user to specify the size of SCM that is desired. The code requires a certain amount of SCM for execution. The default value of MAXSCM is $40{,}000_{10}$ words, a sufficiently large value to handle the majority of one dimensional problems. For the higher dimension solvers this amount may not be enough and the code will inform the user of the amount needed. It should be noted that MAXSCM is the number of words of SCM storage to be allocated, and if greatly overestimated will result in a waste of memory.

Through the use of the input parameter, MAXLCM, in Block-I of the card-image input, the user can specify the maximum amount of large core memory (LCM) he wishes to use. If unspecified, the value of MAXLCM is defaulted to $140{,}000_{10}$ words.

The modular structure of DANTSYS is such that in the processing of each input Block, as well as in the execution of 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 and compared to MAXLCM. If the amount of LCM exceeds MAXLCM, the code will automatically attempt to page LCM information to random disk, such that the stage requires no more than MAXLCM words of LCM at any one time. After Block-I is read, the sum of MAXSCM and MAXLCM is used to obtain an area from the HEAP to be used as the storage for the problem. Again, the code will inform the user of the amount being used and whether information is being paged to disk.

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. Also, if the user is computing on, say, a virtual memory machine with no random disk, the value of MAXLCM must be large enough so that the problem can be run without random disk.

Execution/File Suppression Flags

Included in the Block-I input parameters are several flags which control the execution-flow of the code or the creation of interface files of by the code. These flags are relatively specialized and are normally of interest only to the more advanced user. Accordingly, details on the use of these parameters will not be described here but are provided in another chapter. See "Module Execution Control" on page 13-19.

MORE DETAILS ON GEOMETRY INPUT

Geometry-related information is passed to the appropriate SOLVER module and the EDIT module solely by means of the 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 needs to suppress creation of a new GEODST which would overwrite the existing GEODST file. 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 common to these codes. The terms fine mesh, coarse mesh, and zones are defined below. The term region is not used directly in any DANTSYS input and the user preparing geometry input in Block-II need not be concerned with that term. However, region is a concept used in the GEODST standard file and should the GEODST file be produced from some other code, it is possible that part of the geometry description will be in terms of regions. In any case, DANTSYS will accept any GEODST file that obeys the standard file description.

The fine mesh is the spatial solution-mesh for the problem, as described in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1. Each fine mesh, or fine mesh interval, is bounded by an adjacent pair of fine-mesh grid-lines. In the x direction, these are denoted $x_{i-1/2}$ and $x_{i+1/2}$ with $x_{i-1/2} < x_{i+1/2}$. There are IT, JT, and KT such fine mesh intervals in respectively the x, y, and z directions. No material discontinuities may occur within a fine mesh interval. The specification of the fine mesh is accomplished by specifying how many equally sized fine mesh intervals there are in each 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, JM, and KM coarse mesh intervals spanning the problem in each of the directions. 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. Similarly the arrays YMESH and ZMESH are used for the other directions as appropriate. 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. The input arrays YINTS and ZINTS are used for the other directions. 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 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 geometry input through Block-II card images, the user need not be concerned with the term. However, when the code uses the Block-II information to write a standard GEODST file, 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. In the ZONES array input the zone number, n ($1 \leq n \leq$ NZONE), is determined by the order in which zones are specified in the ASSIGN array input in Block-IV (See "ASSIGNING MATERIALS TO ZONES" on page 11-11), so that the zone number tells the code which macroscopic cross-section set is to be used within that zone. Coarse mesh intervals having the same zone number need not be simply connected.

In the ZONES array, the number 0 (zero) can be used to specify that a coarse mesh interval is a pure void (all cross sections are identically zero). A "0" does not count as a zone in determining the value of NZONE.

The zones array has no meaning in the special case of fine mesh mixing. In that option, the material content of each fine mesh is supplied by the LNK3DNT file.

MORE DETAILS ON SOLVER INPUT

Iteration Strategy

As described in "Iteration Procedure and Diffusion Synthetic Acceleration" on page 12-14 of this document, the ONEDANT, TWODANT, TWODANT/GQ, and THREEDANT solver modules employ the diffusion synthetic method to accelerate the iterative procedure used in solving the transport equation. In this section is described the implementation of the iterative strategy and acceleration method in the solver modules and reflected in the iteration monitor printout supplied as printed output. Also described are the input convergence controls by which the user controls the iteration process.

The basic features of the iteration strategy are shown in the simplified flow diagram of Figure 7.1, "Simplified flow diagram of SOLVER iteration strategy.," on page 7-18. As indicated, there are two different iterative procedures, one for problems containing fissionable material and/or energy-group upscattering 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 within group scattering source and the calculation of the pointwise angular fluxes in each group. 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 a multigroup diffusion calculation where the diffusion coefficient for each space-energy point is evaluated as:

$$D(x,g) = \begin{cases} \dfrac{1}{3\Sigma_{trans}(x,g)} & \text{If } \Sigma_{trans}(x,g) \text{ is available.} \\ & \text{If not,} \\ \dfrac{1}{3\Sigma_t(x,g)} & \text{Isotropic scatter} \\ \dfrac{1}{3\left[\Sigma_t(x,g) - \Sigma_{s1}(x, g \rightarrow g)\right]} & \text{Anisotropic scatter} \end{cases} \quad (1)$$

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, $\Sigma_{trans}(x,g)$ is the macroscopic transport cross section (transport is normally provided on the ISOTXS

or GRUPXS files), 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, be the transport cross section in transport-corrected cross-section libraries for isotropic scatter.

Assuming that the problem contains fissions and the default setting of the input variable, IITL, is taken (IITL=1), then the iteration procedure for solving the transport equation begins with solving the conventional, multigroup diffusion equation:

$$-\nabla \bullet D_g \nabla \phi_g^{k+1} + \sigma_{R,g}(\underline{r})\,\phi_g^{k+1}(\underline{r}) = \frac{\chi_g}{k_{eff}}\Phi^k(\underline{r}) + \sum_{g'=1}^{g-1} \sigma_{s,g' \rightarrow g}\phi_{g'}^{k+1}(\underline{r}) + \sum_{g'=g+1}^{G} \sigma_{s,g' \rightarrow g}\phi_{g'}^{k}(\underline{r}) \qquad (2)$$

$$g = 1, \ldots, G$$

where g is the energy group number,

k is the diffusion sub-outer iteration index,

$\sigma_{R,g} = \sigma_{t,g} - \sigma_{s,g \rightarrow g}$ is the removal cross section,

$\Phi^k(\underline{r}) = \sum_{g'=1}^{G} (\nu\sigma_f)_{g'}\phi_{g'}^k(\underline{r})$ is the fission distribution.

On the right hand side of Eq. (2), the first term is the fission source, the second is the down-scatter source, and the third is the upscatter source. The iteration process starts (k=0) by setting the flux to zero for each group but assuming a spatially flat fission distribution. Equation (2) is then solved by inverting the diffusion operator for each group on the given source, updating the down-scatter source as indicated. The inversion is by line inversion in 1D and by a multigrid procedure in 2- and 3D. Once all the groups are solved, the fission distribution is recalculated using this flux, and the upscatter source is computed if present. This iteration is called a diffusion sub-outer and continues until the convergence criterion is satisfied. This criterion is explained below in Eqs. (6) and (7).

The solution of the transport problem now continues with the inner iteration of each group of the transport equation. This iteration is written as:

$$\Omega \bullet \nabla \psi_g^{l+1/2}(\underline{r}, \Omega) + \sigma_{t,g}(\underline{r})\,\psi_g^{l+1/2}(\underline{r}, \Omega) = \sigma_{s,g \rightarrow g}\phi_g^l(\underline{r}) + \sum_{g'=1}^{g-1} \sigma_{s,g' \rightarrow g}\phi_{g'}^{l+1/2}(\underline{r}) + Q_g(\underline{r}) \qquad (3)$$

$$g = 1, \ldots, G$$

where l is the inner iteration index,

$\psi_g(\underline{r}, \underline{\Omega})$ is the angular flux for group g,

$\phi_g^l(\underline{r}) = \int \psi_g^l(\underline{r}, \underline{\Omega})\, d\underline{\Omega}$ is the transport scalar flux,

$$Q_g(\underline{r}) = \frac{\chi_g}{k_{eff}} \Phi^{CD}(\underline{r}) + \sum_{g'=g+1}^{G} \sigma_{s,g' \to g} \phi_{g'}^{CD}(\underline{r}) \quad \text{source from diffusion,}$$

CD refers to the corrected diffusion calculation.

Equation (3) describes the transport inner with the source fixed from a previous "corrected multigroup diffusion" calculation. The corrected diffusion equation has the same form as the conventional diffusion (Eq. (2) above) except that the diffusion parameters have been changed according to the DSA procedure (see page 12-14 of this document). The scalar flux for the next transport iterate, ϕ_g^{l+1},comes from the solution of the DSA equation for this group, i.e.:

$$-\nabla \bullet \underset{\sim}{D}_g^{l+1/2} \bullet \nabla \phi_g^{l+1} + \sigma_{R,g} \phi_g^{l+1}(\underline{r}) = \sum_{g'=1}^{g-1} \sigma_{s,g' \to g} \phi_{g'}^{l+1}(\underline{r}) + Q_g(\underline{r}) \qquad (4)$$

This is written for the diffusion correction method; see page page 12-14 for the definition. In the default transport iteration mode, only one iteration of Eqs. (3) and (4) is done per group until the fission and upscatter sources have converged (this convergence is described in the section: transport source iteration convergence, below). Once each group has processed one inner iteration, the parameters for the corrected, multigroup diffusion equation have been generated as well as the transport scalar fluxes, $\phi_g^{l+1}(\underline{r})$. These are then used to start another multigroup diffusion iterative solution analogous to Eq. (2) except with the transport corrected diffusion equation (again see Eq. (11) on page 12-16). This process defines the transport outer iteration which continues until the fission and upscatter sources have converged. Once this convergence has been obtained, then IITL is set to IITM and the inner iteration process for each group is continued until the scalar flux has converged (see inner iteration convergence, below), or until IITM is attained, whichever comes first. This usually completes the iteration process of the transport solution, although, it is possible that converging the inners has caused a change in the fission source that exceeds the convergence criterion. If this is so then another transport outer is performed.

For problems that contain neither fission nor upscatter, the transport outers are not done, IITL is defaulted to IITM, and only the inner iteration DSA convergence procedure is done.

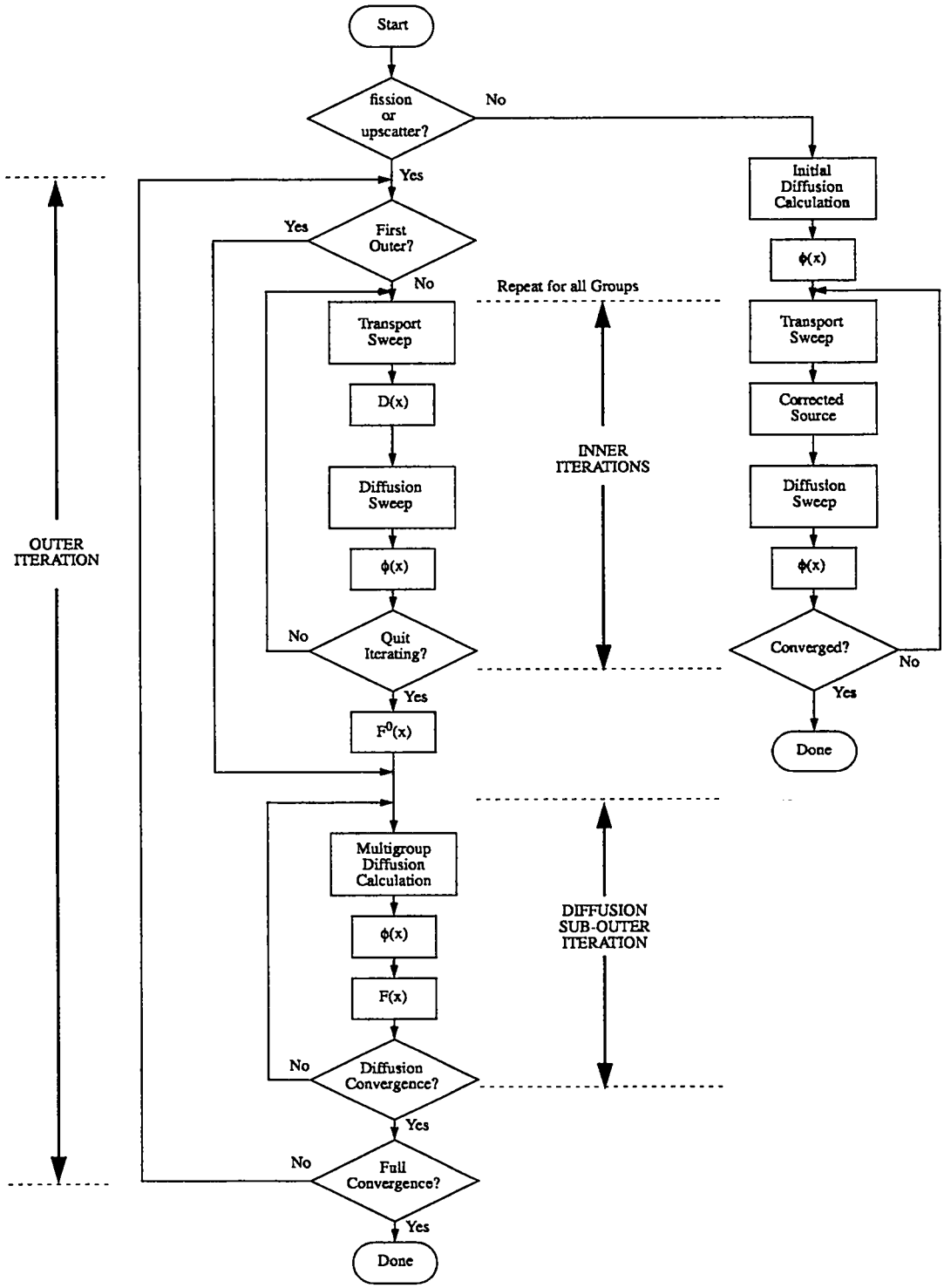


Figure 7.1 Simplified flow diagram of SOLVER iteration strategy.

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.

Inner Iteration Convergence.

The inner iterations for a given energy group are said to be converged when the point-wise scalar fluxes from one inner iteration to the next satisfy the condition:

$$max \left| \frac{\left(\phi_{i,g}^{l+1} - \phi_{i,g}^{l} \right)}{\phi_{i,g}^{l}} \right| < EPSI \ , \tag{5}$$

where $\phi_{i,g}^{l}$ 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.

Diffusion Sub-Outer Iteration Convergence.

The convergence of the diffusion sub-outer iterations requires the satisfaction of two criteria. We use the index k to denote the diffusion sub-outer iteration number, p to denote the transport outer, g to denote the group number, and i the spatial point. Convergence of the diffusion sub-outers is then satisfied when both

$$\max_{i} \left| \frac{\left(\Phi_{i}^{k+1} - \Phi_{i}^{k} \right)}{\Phi_{i}^{k}} \right| < 3.0*EPS \tag{6}$$

and

$$\left| 1 - \lambda_{D}^{k+1} \right| < \mathrm{EPS} \ . \tag{7}$$

where

Φ^{k} is the fission distribution at sub-outer k,

$EPS = max\,(EPSI, 0.01*(0.1**\mathrm{p})\ )$,

$$\lambda_D^{k+1} \equiv \frac{\left(\Phi^{k+1}, 1\right)}{\left(\Phi^{k}, 1\right)} .$$

The notation (F,G) denotes the inner product, or volume integral, of the product F*G. In the above, λ_D^{k+1}, measures the precision of the k eigenvalue. Note that the definition of EPS involves a strategy where the diffusion solution convergence depends upon the transport outer iteration, eventually attaining the precision EPSI.

Transport Source Iteration Convergence.

The convergence of the transport source triggers the increase of the number of transport inners from IITL=1 to IITL=IITM if the default strategy is taken. We measure the precision of the transport fission source by comparing the change from the end of the diffusion sub-outers to that resulting from the next transport inners. That is the fission source is deemed sufficiently converged if

$$\max_i \left| \frac{\left(\Phi_i^{p*} - \Phi_i^{p}\right)}{\Phi_i^{p}} \right| < 10.0*EPSI \qquad (8)$$

and

$$\left|1 - \lambda^{p*}\right| < 1.5*\text{EPSI} , \qquad (9)$$

where p* denotes the fission distribution evaluated from the transport scalar flux resulting from the transport inners after the completion of transport outer p (which is the completion of the solution of the DSA multigroup diffusion equation). Thus,

$$\lambda^{p*} \equiv \frac{\left(\Phi^{p*}, 1\right)}{\left(\Phi^{p}, 1\right)} . \qquad (10)$$

Final Convergence

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 no upscatter and no fission, 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 (IEVT > 0), the default 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 20-50 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. Final convergence is obtained when the change in the fission distribution and the eigenvalue is less than EPSI, i.e., when

$$\left|1 - \lambda^{p+1}\right| < \text{EPSI} \ ,$$

and

$$\max_i \left| \frac{\left(\Phi_i^{p+1} - \Phi_i^p \right)}{\Phi_i^p} \right| < EPSI \tag{11}$$

These outer iterations are thus terminated when either the above 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.

Grey Acceleration of Upscatter

In ONEDANT and TWODANT there also exists a grey accelerator for upscatter problems. This accelerator is of some help when the number of upscatter groups exceeds 6 or so, the materials exist in large regions and are "good" scatterers such as deuterium, graphite, water, etc. The main idea is to collapse the upscatter groups into one group (hence the name grey) and solve the one-group DSA equation for that group. From this is generated a correction to the flux in the upscatter region which can at times greatly accelerate the convergence of the upscatter iteration. This option is invoked by setting the Block-V input parameter, GREYACC, to "yes" or to "1."

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.

General Aspects of the Monitor Print.

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 criteria of Eqs. (6) and (7) 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. (5). Finally are included the values of $\lambda^p - 1$ and the maximum pointwise flux error corresponding to the values used in the test for full convergence given by Eq. (11).

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 normally provided for a fixed-source problem without fission or upscatter (IEVT= 0) since only one outer iteration is required. The inner iteration monitor print is triggered by IITL being equal to IITM; this condition is assured by the defaults for these quantities when IEVT=0. The user may thus suppress this portion of the print by setting IITL to some number different from IITM. For other problems (IEVT≠ 0) the inner iteration monitor is only provided for outer iterations following the satisfaction of the "nearly converged" conditions of Eqs. (8) and (9) at which time IITL is set equal to IITM by the code. In the inner iteration monitor are included the group number; the number of inner iterations taken, the maximum pointwise scalar flux error (see Eq. *), and the spatial mesh point where this maximum error occurred.

Warning Messages and Their Meanings

In the inner iteration monitor several warning messages are provided for the user if the calculation encountered some difficulty.

A message "ACCELERATION DISABLED"' is printed 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 accelerated.

The message "TRANSPORT FLUXES BAD" is a more serious warning. It 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 fatal, it does usually indicate that the spatial mesh is too coarse and that the results are suspect.

The message "NEG. SOURCE" is printed when a negative angular source term is calculated for one or more mesh cells (ONEDANT only). This situation can only occur when an anisotropic scattering source is being used, that is, when the parameter ISCT in Block-V is greater than zero. A negative source for a given angular direction is usually the result of the truncation of the Legendre expansion of the scattering source term in mesh cells where scattering dominates the source term. Both the scattering cross sections and the angular flux must be quite anisotropic for this condition to occur.

If the message "NEG. SOURCE - ACCELERATION DISABLED" appears, it means that the presence of negative angular sources has occurred to such a level that the syn-

*

thetic diffusion equation could not be used to accelerate that inner iteration. The presence of this message does not necessarily make the answers suspect if convergence is achieved; it merely tells the user that the inner iteration could not be accelerated.

The message "NEG. SOURCE - TRANSPORT FLUXES BAD" is the most serious ramification of the presence of negative angular sources. It means that nonpositive transport scalar fluxes existed following the last inner iteration in conjunction with negative angular sources. Nonpositive scalar fluxes are not necessarily fatal, but they usually indicate that the truncated Legendre scattering expansion was quite poor and that any results are suspect.

The possible remedy for any of the "NEG. SOURCE" occurrences is to increase the Legendre scattering expansion order (increase the value of ISCT in Block-V) if the cross-section library contains data for the higher order scattering. The other remedy is to use the Cesaro transport correction described in "Transport Corrections for the Cross Sections (TRCOR)" on page 7-31.

Boundary Conditions

Several boundary condition options are available to the user 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, that is, in the one dimensional case, for example,

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

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 $\mu = 0$.

Additional input parameters IBT and IBB are used for the top and bottom boundaries of two-dimensional problems and IBFRNT and IBBACK are used for the front and back of three-dimensional problems.

In the ONEDANT module, 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 card-image input.

- Surface source boundary conditions -- the incoming angular fluxes on the boundary are set equal to user-supplied values in Block-V. See "Input of Inhomogeneous Sources" on page 7-26.

Input of Quadrature Sets

The DANTSYS code package has the option of obtaining the discrete-ordinates angular quadrature coefficients from a SNCONS standard interface file[1], from one of three 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 12.4, "Number of Quadrature Points, M as a Function of Sn Order, N" on page 12-27.

The built-in constants provided in the ONEDANT module 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_2, S_4, S_6, 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 fully symmetric quadrature set.

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 "Spherical Harmonics Expansion of the Scattering Source" on page 12-22 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 Figure 12.4, Figure 12.5, and Figure 12.6. 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 regardless of the value of IQUAD entered in the input, that is, the WGT and MU input arrays will override any other source of quadrature constants.

For the two and three dimensional modules, the default quadrature set is the so called TWOTRAN set which goes up to S_{16} in even orders. This is a triangular set which is based on Gaussian levels measured from the y axis in xy geometry or the z axis in rz and the three dimensional cases. An attempt is made to make this set as symmetric as possible to rotations about the coordinate axes. There is also available a Gauss-Chebychev set triggered by the Block-V input variable IQUAD =|2|. This set exists up to S_{50} in even values of S_n order and up to S_{100} in steps of 10 from S_{50}. The main attribute of this set is that it exactly integrates the spherical harmonics up to order Sn-1. For more information, see Ref. 3.

Zone-Dependent Fission Fractions (the CHI Array)

For problems containing fissile or fissionable nuclides it is necessary to provide the code with the groupwise fission fractions, χ_g, g=1,2 . . . , NGROUP. From certain of the cross-section libraries (ISOTXS, GRUPXS, MENDF, etc.) a single χ vector characterizing the dominant fissionable isotope (U235, U238, Pu239, etc.) can be extracted automatically from the library and used for the problem. Alternatively, a single χ vector can be provided by the user in Block-III through the card-image CHIVEC input array. Again, this is a single χ vector characterizing the dominant fissionable isotope in the problem.

In Block-V of the input is provided the card-image CHI input array that can optionally be used to provide multiple χ vectors so that each zone in the problem can use the fission fraction characteristic of the dominant fissionable isotope in that zone. The availability of such zone-dependent χ's is particularly useful for problems in which one zone may have a different dominant fissile isotope than another zone. The CHI array is input as N distinct χ vectors ($1 \leq N \leq$ NZONE) with each vector containing NGROUP χ values. The first χ vector is applied to the first zone, where the first zone is that defined by the first entry in the ASSIGN array of Block-IV; the second χ vector is applied to zone number 2 (defined by the second entry in the ASSIGN array), etc. If the total number of χ vectors in the CHI array is less than NZONE (the total number of zones), then the last χ vector in the CHI array is used for all remaining but unspecified zones.

Note that if the CHI array is used in Block-V, its entries will override the χ's provided either from the cross-section library or from the CHIVEC array in Block-III.

Input of Inhomogeneous Sources

Any of the solver modules of DANTSYS will solve the inhomogeneous transport equation in the multigroup, discrete-ordinates approximation form, using the method outlined in "Iteration Strategy" on page 7-15. The user specifies this type of calculation by setting the input control word IEVT to 0 or -1 for ONEDANT and 0 for all the other solvers. In ONEDANT, IEVT=0 is used when there is neither fissionable material nor upscattering in the problem and IEVT= -1 is used when either fissionable material is present (but not in sufficient amount to make the system nuclearly critical or supercritical) or upscattering is present.

The user must supply the specifications for the inhomogeneous sources either in the input or from a FIXSRC[3] 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 FIXSRC standard interface file, the user sets the input control word INSORS to 1. 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.

Distributed Source Input

As described in "Spherical Harmonics Expansion of the Inhomogeneous Source" on page 12-26, the inhomogeneous distributed source must be represented by the spherical harmonic expansion in multigroup form (from Eq. (20) on page 12-26) :

$$Q_g(r,\Omega) = \sum_{n=1}^{NMQ} (2L+1)\, R_n(\Omega)\, \tilde{Q}_{n,g}(r) \quad , \; g = 1, \ldots, NGROUP \qquad (12)$$

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. (12). 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 $(r,n) = \tilde{Q}_{n,g}(r)$. There are corresponding source distribution vectors in the y and z directions named SOURCEY and SOURCEZ, respectively. The number of moments, NMQ, in Eq. (12) 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 12.2, "Number of Spherical Harmonics, N, as a Function of Order" for each geometry. For example, if one wishes to enter a P_3 inhomogeneous source in cylindrical geometry, Table 12.2 shows that six spherical harmonics are required for P_3 in cylindrical geometry. Table 12.3 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 Eq. (19a) on page 12-26 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:

$$\tilde{Q}_{1,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi P_0(\xi)\ Q_g(r,\xi,\phi) = Q_{0,g}(r)\ , \qquad (13\text{ a})$$

$$\tilde{Q}_{2,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi P_1^1(\xi)\cos\phi\ Q_g(r,\xi,\phi) = Q^1_{c,1,g}(r)\ , \qquad (13\text{ b})$$

$$\tilde{Q}_{3,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi P_2(\xi)\ Q_g(r,\xi,\phi) = Q_{2,g}(r)\ , \qquad (13\text{ c})$$

$$Q_{4,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi \frac{\sqrt{3}}{6} P_2^2(\xi)\cos 2\phi\ Q_g(r,\xi,\phi) = Q^2_{c,2,g}(r)\ , \qquad (13\text{ d})$$

$$\tilde{Q}_{5,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi \frac{\sqrt{6}}{6} P_3^1(\xi)\cos\phi\ Q_g(r,\xi,\phi) = Q^1_{c,3,g}(r)\ , \qquad (13\text{ e})$$

and

$$\tilde{Q}_{6,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi \frac{\sqrt{10}}{60} P_3^3(\xi)\cos 3\phi\ Q_g(r,\xi,\phi) = Q^3_{c,3,g}(r)\ , \qquad (13\text{ f})$$

for n=1, . . , 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, . . . , IT.

It is worth noting that most inhomogeneous distributed sources are assumed to be isotropic, so that NMQ in Eq. (12) is unity and the only source moment entered is the zeroth moment.

$$\tilde{Q}_{1,g}(r) = \frac{1}{4\pi}\int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi Q_g(r,\xi,\phi) = Q_{0,g}(r)\ , \qquad (14)$$

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).

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:

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

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

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] 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 external boundaries of the physical problem and not at internal interfaces. For the angle-dependent surface sources, only the incoming directions are required, and they must be ordered as described below. There are corresponding arrays for the top and bottom boundaries of two-dimensional problems and for the front and back of three-dimensional problems.

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 at the time of code execution.

Angle-dependent surface sources can be input through the SARITE and SALEFT input arrays. Currently, SALEFT can only be used in plane (X, XY, XYZ) geometry. The surface sources are actually angular fluxes, $\psi_{m,g}$, at the right or left surfaces [m denotes the order index (with m correlated to specific quadrature directions), g denotes energy group]. There are corresponding arrays for the two and three dimensional geometries (SABOTT, SATOP, SAFRNT, SABACK). In ONEDANT, the order of SARITE and SALEFT entries is as follows:

$$\psi_{1,1}\ , \psi_{2,1}\ \psi_{3,1}, \cdots; \psi_{1,2}, \psi_{2,2}, \psi_{3,2}, \cdots ; \cdots$$

- Slab Geometry (IGEOM= 1 or SLAB or PLANE):

The ordering of the angles is as shown below (S_6 used for illustration).

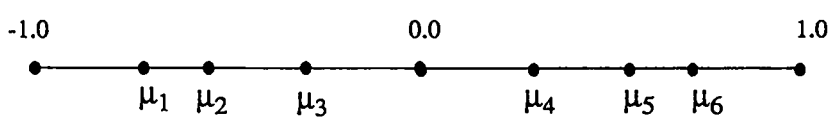


Figure 7.2 Ordering in slab geometry.

with $|\mu_1| = \mu_6$, $|\mu_2| = \mu_5$, and $|\mu_3| = \mu_4$.

Using the S_6 quadrature illustrated above, the correspondence between the SARITE and SALEFT ordering index, m, and the angular directions (shown above) is as follows:

Table 7.1 Source Ordering Index in Slab Geometry.

	SARITE	SALEFT
m	μ	μ
1	μ_1	μ_4
2	μ_2	μ_5
3	μ_3	μ_6

NOTE: The SALEFT ordering differs from the SARITE ordering in terms of the $|\mu|$ associated with the ordered angular fluxes. For example, the first entry in SARITE (for each group) corresponds to μ_1 while the first entry in SALEFT (for each group) corresponds to μ_4 and $|\mu_1| \neq \mu_4$.

- Cylindrical Geometry (IGEOM= 2 or CYLINDER or CYL):

Using S_6 for illustrative purposes, the principal octant ($\mu > 0, \eta > 0$) of the unit sphere of angular directions is as shown below.

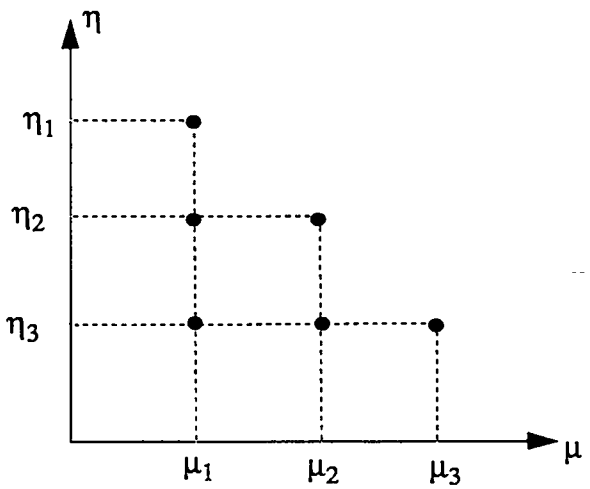


Figure 7.3 Quadrature points in cylindrical geometry.

The order, m, in which the SARITE angular fluxes are entered for each energy group is correlated to the (μ,η) angular directions (shown above) in the table below:

Table 7.2 Source Ordering Index in Cylindrical Geometry

m	μ	η
1	$-\mu_1$	η_1
2	$-\mu_2$	η_2
3	$-\mu_1$	η_2
4	$-\mu_3$	η_3
5	$-\mu_2$	η_3
6	$-\mu_1$	η_3

There are corresponding angular flux arrays for the top and bottom boundaries of two-dimensional problems and for the front and back of three-dimensional problems.

Normalization of the Calculation (the NORM Parameter)

In Block-V of the input is the parameter NORM. Through the use of NORM the user may specify whether or not the calculation should be normalized to a particular particle production rate as described below.

If NORM= 0 or omitted from Block-V no normalization is done.

If NORM= XX, where XX is positive, and inhomogeneous (fixed) sources exist (IEVT $\geq$ 0), then the calculation will be conducted and normalized such that the total rate at which particles are introduced into the system from inhomogeneous sources is XX. Thus, all fluxes that are in the calculation are normalized to the values they would have if the total rate at which particles are introduced in the system from inhomogeneous sources (distributed and/or surface sources) is XX particles/unit time (or particles if fluxes are to be interpreted as fluences).

If NORM= XX where XX has a positive value, and an eigenvalue calculation is being conducted (IEVT > 0), then the calculation is performed and normalized such that the total rate at which neutrons are produced by fission reactions is XX. Thus, all fluxes that are computed in the calculation are normalized to the values they would have if the total rate at which neutrons are produced by fissions in the system is XX neutrons/unit time (or neutrons if fluxes are to be interpreted as fluences).

Note that NORM applies normalization only to particle production rates. Normalization of edit results to a desired fission power level can be done through the use of the POWER and MEVPER parameters in Block-VI as described in "Power Normalization" on page 2-64, page 3-71, page 4-74, page 5-56, or page 6-68. These latter two parameters effect normalization in only the Edit Module.

Transport Corrections for the Cross Sections (TRCOR)

In Block-V of the input is the parameter TRCOR which permits the user to have the solver module perform its calculation using transport-corrected cross sections. Below is provided a little background on the use of the transport correction in DANTSYS.

As described in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual", a truncated spherical harmonics, or Legendre polynomial, expansion of the scattering sources is made per Eq. (14) there. The scattering order ISCT (an input parameter in Block-V) determines where the expansion is to be truncated. Any value of ISCT ($0 \leq$ ISCT $\leq$ MAXORD) is allowed. Recall that MAXORD is the highest Legendre order for which scattering matrix cross sections exist in the cross-section library being used. The principal reason for choosing to truncate the expansion to an order less than MAXORD is because of computer storage. Each term in the spherical harmonics (Legendre) expansion requires computing and storing an angular flux moment for every energy group and spatial mesh cell, and the number of such flux moments increases with scattering order (see Table 12.2, "Number of Spherical Harmonics, N, as a Function of Order" on page 12-24).

If ISCT is less than MAXORD, it is usually advantageous to apply a transport correction to the truncated Legendre scattering cross sections. Transport corrections are designed to account approximately for terms in the expansion being omitted by the truncation. There are three different types of transport corrections allowed in the code. They are selected through the use of the parameter TRCOR in Block-V of the input.

- If TRCOR= NO or if TRCOR is omitted, no transport correction is applied.

- If TRCOR= DIAG the diagonal transport correction is applied. With this correction the material macroscopic total cross section for each group and the material macroscopic within-group scattering cross section for each group and each scattering order is modified as shown below:

$$\hat{\sigma}_{t,g} = \sigma_{t,g} - \sigma_s^{ISCT+1}(g \rightarrow g)$$

and

$$\hat{\sigma}_s^l(g \rightarrow g) = \sigma_s^l(g \rightarrow g) - \sigma_s^{ISCT+1}(g \rightarrow g),\ l = 0, 1, \ldots, ISCT .$$

where the notation $\hat{\sigma}$ denotes "transport corrected" cross section. The diagonal transport correction is normally recommended.

- If TRCOR=BHS, the Bell-Hansen-Sandmeier transport correction is applied. With this correction the macroscopic total cross section for each material and group and the macroscopic within-group scattering cross sections for each scattering order, material, and group are modified as follows:

$$\hat{\sigma}_{t,g} = \sigma_{t,g} - \sum_{g'} \sigma_s^{ISCAT+1}(g \rightarrow g')$$

and

$$\hat{\sigma}_s^l(g \rightarrow g) = \sigma_s^l(g \rightarrow g) - \sum_{g'} \sigma_s^{ISCAT+1}(g \rightarrow g'),\ l = 0, 1 \ldots, \text{ISCT} .$$

- If TRCOR=CESARO, the n^{th}-Cesàro-mean-of-order 2 transport correction is applied where n=ISCT. With this correction the macroscopic group-to-group scattering cross section for each scattering order, zone, and group is modified as shown below:

$$\hat{\sigma}_s^l(g \rightarrow g') = \frac{(ISCT+2-l)\ (ISCT+1-l)}{(ISCT+2)\ (ISCT+1)} \sigma_s^l(g \rightarrow g'),\ l = 1, \ldots, ISCT .$$

This Cesàro correction ensures that the truncated Legendre expansion for the scattering cross section from group g to group g´ is (i) positive, (ii) preserves the $l = 0$ (i.e., P_0) moment of the scattering cross section, and (iii) converges to the same value as $\sigma_s^l(g \rightarrow g')$ for large values of ISCT (see Ref. 2).

Note that if ISCT= 0, the Cesàro correction does nothing to the cross sections -- the multiplier of $\sigma_s^l(g \rightarrow g')$ is unity. Generally, the Cesàro transport correction should only be used with ISCT ≥ 2.

At the time of the writing of this manual, little experience has been had with the Cesàro correction in terms of its accuracy. As a result the user is cautioned regarding its use. However, using this transport correction will eliminate the possibility of negative sources, as discussed in "Warning Messages and Their Meanings" on page 7-22.

Please note that the transport corrections described above are only applied to the macroscopic cross sections used in the solver module. None of the cross-section files are altered by the use of the transport correction in the solver module.

Buckling Corrections

Leakage from the transverse dimension(s) of a multidimensional system may be simulated in a lower dimensional solver by using a user-specified buckling height (BHGT) and/or buckling width (BWTH) in the Block-V card-image input. For plane and 2-angle plane geometries (IGEOM=1) in ONEDANT, both BHGT and BWTH may be specified. For cylindrical geometry only the buckling height, BHGT, may be specified. For TWODANT, only BHGT may be used. The buckling dimensions are in units consistent with the units on cross section, for example, in cm if macroscopic cross sections are in cm^{-1}. If diffusion theory is assumed adequate, then the flux shape in the transverse directions, 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 given by

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

where σ is the macroscopic zone total cross section, BHGT (or similarly BWTH) is the buckling height (or buckling width), and 1.4209/σ 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.

Eigenvalue Searches

It is possible 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:

IEVT[a]	Type of Eigenvalue Search
2	Time absorption (alpha)
3	Concentration
4	Critical size (dimension)

a. Not included here are the options IEVT=-1 for inhomogeneous source problems with upscatter and/or fission, IEVT=0 for inhomogeneous source problems without upscatter or fissions, 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 supercritical 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 "Mixing Array for a Concentration Search" on page 2-44, page 3-46, or page 6-46).

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

$$\tilde{R}_{k+1} = \tilde{R}_k + (R_{k+1} - R_k) * (1 + EV * RM_k), \quad k=1,2,\ldots, IM \ , \qquad (15)$$

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 inter-

face 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

$$\lambda = \frac{(\text{Fission source})^{k} + \text{Inhomogeneous source}}{(\text{Fission source})^{k-1} + \text{Inhomogeneous source}} \qquad (16)$$

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 Figure 7.4 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 EPSO. 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 equal to 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.

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 λ - 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, a linear fit is used and the new eigenvalue is computed from

$$(EV)_{new} = (EV)_{old} + POD * EVS * (1 - \lambda) \quad , \tag{17}$$

where POD is an input "parameter oscillation damper" that may be used to restrict the amount of change in the eigenvalue. In Eq. (17), EVS is a measure of the slope of the curve. When $|1 - \lambda| > XLAH$, $(1 - \lambda)$ in Eq. (17) 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.

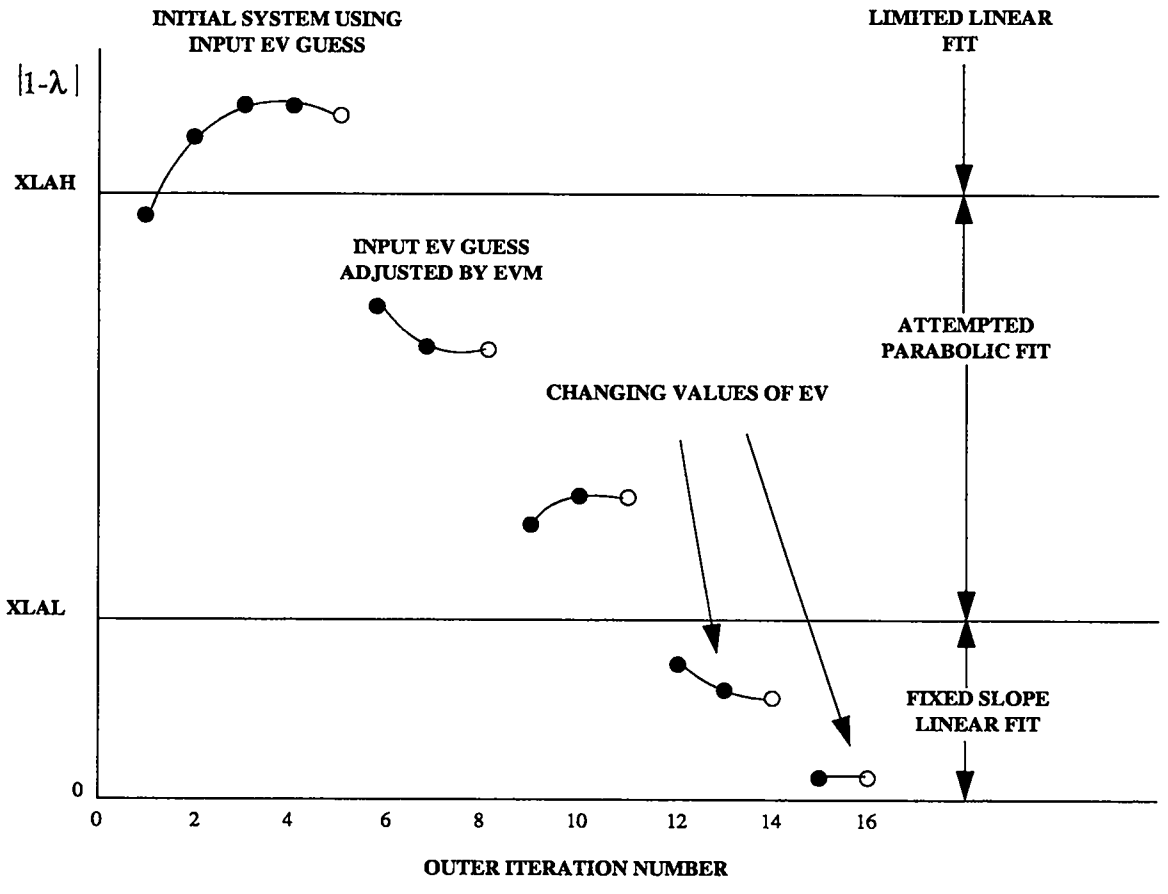


Figure 7.4 Variation of λ during a hypothetical eigenvalue search.

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

Adjoint Computations

The DANTSYS solver modules solve 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 $\chi_g \Sigma (\nu\sigma_f)_h \phi_h$, one has $(\nu\sigma_f)_g \Sigma \chi_h \phi_h$. 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 Ω must be identified as the adjoint solution for direction Ω. 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[1] standard interface file.

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 "INTERFACE FILES USED IN MIXING" on page 11-17, 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 "Adjoint Edits" on page 8-15.

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. T. E. Albert and P. Nelson, "Computation of Azimuthally Dependent Albedo Data by Invariant Embedding," in Proc. of Sixth Intl. Conf. on Radiation Shielding, May 16-20, 1983, Tokyo, Vol. I, pp. 283-293.
3. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).

RUNNING THE EDIT MODULE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

8

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

The basic function of the Edit Module is to perform postprocessing, or edit, operations after the flux calculation is done. Two basically different types of edits may be done: a simple inventory of the masses in the problem, and a calculation of reaction rates using the multigroup, pointwise scalar fluxes generated by the execution of the solver module or, perhaps, by 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.

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 Table 13.1, "Files Read and Written," on page 13-9). Included in these general files are the geometry specifications (GEODST file), the material mixing and cross-section specifications (NDXSRF, ZNATDN, and SNEXDT files), and the assignment of materials to zones specifications (ASGMAT file). 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 whenever 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 DANTSYS 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 written, 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).

REACTION RATES

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

$$\rho_{i,g'} = \sum_{g \epsilon g'} \phi_{i,g} R_{i,g} \quad , \tag{18}$$

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 "Energy Group Options For Edits" on page 8-10) 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).

In three-dimensional problems a special point edit by planes has been included. Input specifications in Block-VI includes the arrays IPLANE, JPLANE, and KPLANE. These are a set of integers which designate which planes of point edits will be printed. If any of these arrays are specified, then the POINTS array must not be specified. Thus, to get point edits of all the points on the x-y planes, 1, 5, and 10, for example, set KPLANE = 1,5,10.

To interface with a specific graphics package, TECPLOT© which is commercially available, the edit module also has the capability to generate TECPLOT files which mirror the printed output above. This capability is avialable for 2- and 3-dimensional geometries only. The generation of these files and the printing of the point edits is controlled by the input parameter PRPLTED in Block-VI. Thus, PRPLTED = 0/1/2/3 is chosen to print only / no point edit output / TECPLOT files only / both generate the TECPLOT files and print the point edits.

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 ONEDANT 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 shown in the chapter "FREE FIELD INPUT REFERENCE," 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, and 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

$$\rho_{Z_{m,g'}} = \sum_{g \varepsilon g'} \sum_{i \varepsilon Z_m} \phi_{i,g} R_{i,g} V_i \ . \tag{19}$$

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 XMESH input array in Block-II). An exception is TWODANT/GQ, for which the default is the solver zones.

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

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 quantities be calculated as integrals (sums) over the three that edit 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.

Forms Of Response Functions

As indicated in the preceding sections, reaction rate 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.

Cross-Section Response Functions: EDXS

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 the table "Edit Cross-Section Types by Position and Name" on page 2-60 or "MENDF Library Edit Cross Sections" on page 2-66. 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 character names as given in the same tables.

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 the table on page 2-66, 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.

1. 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 as the response function, namely

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

   the Block-VI input parameter RESDNT is set to unity.

2. 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 SNEXDT, which themselves originally came from the basic cross-section 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.

3. 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.

4. 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.

User-Input Response Functions: RSFE, RSFX

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 for a one-dimensional case. 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 (It is defaulted to unity everywhere). There are RSFY and RSFZ input arrays to specify the distribution in the second and third dimensions, if needed. These arrays are also defaulted to unity everywhere.

The RSFE input array can be used 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.

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.

Cross-Section Response Functions Sums: MICSUM

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 character names (from the basic isotope input library) is given followed by a

set of cross-section type position numbers of character names (See "Edit Cross-Section Types by Position and Name" on page 2-60). 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 = PU239, PU240, PU241, U238,

and the MICSUM array were specified as

MICSUM= PU239, PU241, 0, "N-GAMM," "N-FISS," 0,
PU240, U238, 0, ABS

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

$$\text{SUM 1:} \left\{ \left[N_i (\sigma^{n,\gamma} + \sigma^f)_g \right]^{PU239} + \left[N_i (\sigma^{n,\gamma} + \sigma^f)_g \right]^{PU241} \right\} * \phi_{i,g}$$

$$\text{SUM 2:} \left[(N_i \sigma_g^a)^{PU240} + (N_i \sigma_g^a)^{U238} \right] * \phi_{i,g} \ .$$

User-Input Response Functions Sums: IRSUMS

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.

Adjoint Edits

The EDIT module will normally perform regular (forward) edits using regular (forward) scalar fluxes from an RTFLUX standard interface file. If adjoint edits are to be performed, the user need only set the Block-VI input parameter AJED to unity and ensure that the appropriate adjoint scalar flux file (ATFLUX file) exists and is available to the EDIT module at the time of execution. ("Adjoint Computations" on page 7-37 describes adjoint calculations in the SOLVER module.) In the adjoint mode, the EDIT module performs all adjoint reversals and provides the correct group ordering in the printed output.

ASCII File Output Capabilities (the EDOUTF Parameter)

Because of the desirability of presenting output quantities from a calculation in graphics form (plots, tables, etc.), it is valuable to have output in eye-readable ASCII files. Such files are easy to use for extracting information and data and putting it in forms usable by graphics packages, text-editors, etc. The Edit Module can optionally create two such ASCII files for the user. The EDTOUT file is an ASCII file which contains certain geometric information (such as problem geometry, number of coarse mesh intervals, the coarse mesh boundary positions, and number of fine mesh intervals per coarse mesh) together with the edit quantities produced by the Edit Module. The EDTOGX file is an ASCII file containing geometric information, group-total fission source (if problem had fissionable material), and, optionally, the multigroup scalar fluxes at each fine mesh point. File descriptions for these two files are provided in the chapter "FILE DESCRIPTIONS" starting on page 15-1.

The Edit Module (Block-VI) parameter EDOUTF controls the creation of the EDTOUT and EDTOGX files as described in "ASCII File Output Capabilities (the EDOUTF Parameter)" on page 8-15.

MASS INVENTORIES

When isotopic atomic weights are available, the EDIT Module can, if requested by the user, calculate the mass inventories in the problem.* The default is to calculate the inventory of each isotope in each of the zones used by the SOLVER Module. Optionally, mass inventories for each of the EDIT zones may be calculated.

The input variable for the mass edit in Block-VI is "MASSED." The allowed input values are shown in Table 8.1.

Table 8.1 MASSED Input Values

MASSED	Action
0	none
1[a]	Edit by Solver zones
2	Edit by Edit zones[b]
3	Edit by both Solver and Edit zones

a. One is the default if there is any other Block-VI input. If the block is empty, then zero is the default.

b. For all of the DANTSYS codes except TWODANT/GQ, the default for the edit zones is the coarse mesh. For TWODANT/GQ, the default is the solver zones.

The isotopic atomic weights must be available to the Edit Module for the mass edit to work. They may be input in the Block-IV ATWT array, or they may be present in the cross-section library file. In either event, the atomic weights are written to the NDXSRF standard interface file, from whence the Edit Module acquires them.

*If the problem geometry has an infinite lateral dimension, the masses (and volumes) are given per unit height.

For problems in which the fine mesh mixing option is used, the MASSED input values are changed slightly as shown in Table 8.2.

Table 8.2 MASSED Input Values for Fine Mesh Mix Problem

MASSED	Action
0	none
1[a]	Total inventory for the whole problem
2	Inventories by Edit zones[b]
3	Inventories for both Total and Edit zones

a. One is the default if there is any other Block-VI input. If the block is empty, then zero is the default.

b. For linked problems, a Block-VI EDZONE array is required since there is no coarse mesh default for the edit zones. The coarse mesh and the fine mesh are identical in linked problems.

FREE FIELD INPUT REFERENCE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

9

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

This chapter serves as the reference manual for the free field input format as implemented in DANTSYS™. First is a section defining terms and concepts. That is followed by a detailed reference section for the input rules, syntax, and operators for the free field input.

Lastly is a description of the alternate, fixed field, FIDO format. The alternate, fixed field FIDO format is an option for card input cross sections in DANTSYS.

DANTSYS is a trademark of the Regents of the University of California, Los Alamos National Laboratory.

FREE FIELD DEFINITIONS

There are four basic input quantities in the free field input; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array."

The favored input form (there are several) is one very similar to NAMELIST. Each input array has a unique name. To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, could look like:

```
SOURCX= 0 0 0 1.1 1.1 0 0 0 0 0
```

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Unlike NAMELIST, however, an array name CANNOT use a subscript. The operator S, described later, may be used for this addressing function.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Numeric Data Items

Numeric data items follow a FORTRAN input convention. For example, all of the following are valid entries for the number ten:

```
10, 1.0+1, 1E1, 10.0
```

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a FORTRAN variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters or blanks may be included if the data item is surrounded by double quotes. Operators may NOT be used with character data items.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this; the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read, no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input. Most of these operators are FIDO[1] operators, but several are new and represent extensions to FIDO.

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

n *O* d

where:

n is the "data numerator," an integer, or a blank;
O is any one of several "data operators" described in Table 9.1; and
d is a "data entry" (may be blank for some operators).

Note: The "data operator" character must be appended to the "data numerator."

Using operators, the SOURCX input described above could more succinctly be given as:

```
SOURCX= 3r 0 2r 1.1 f0
```

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO operators may be used in numeric entry.

A complete list of the valid operators is given in Table 9.1, "Free Field Data Operators," on page 9-15.

FREE FIELD INPUT DETAILS

This section describes the various rules, restrictions, and options available to the user when creating the input for DANTSYS. First are described the details associated with free-field form of input. Next is presented the information needed for user-specified input forms. This is followed by information for fixed-field FIDO input.

Throughout this section we will use the term *card* or *card-image* to denote a single record or line of characters. Thus, an 80-column card (or card-image) refers to a line containing 80 columns into which ASCII characters may be written one per column.

Free-Field Input

1. Card-Image Ground Rules
   a. Eighty (80) columns available.
   b. No special columns; that is, 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, for example, 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. Character Data Item Ground Rules

   a. Must begin with alphabetic character [see c. below for exception] and may contain from one to eight characters.

   b. Must not contain any of the following characters: =, $, *, blank, comma, slash, semicolon, double quote ("). [See c. below for exception.]

   c. Character data words may be delimited with double quotes to override a. and b. restrictions. For example, PU/239 is not allowed, but "PU/239" is allowed. Remember the eight character limit.

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, for example, 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 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 character data items.

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

   n *O* *d*

   where:

   n is the "data numerator," an integer, or blank;
   O is any one of several "data operators" described in Table 9.1; 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.

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.

Table 9.1 Free Field Data Operators

Data Numerator	Data Operator	Data Entry	Remarks
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 n interpolated entries equally spaced between d and the next data entry. 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	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 as 1 2 3 2Q3.

Table 9.1 Free Field Data Operators (Cont.)

Data Numerator	Data Operator	Data Entry	Remarks
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. For example, 4 5 6 6 5 4 4 5 6 can be input as 4 5 6 2N3.
n	M	m	SEQUENCE REPEAT OPTION 4: Same effect as "N" except the sign of each entry in the sequence is reversed each time the sequence is entered. 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 m strings 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.)
n	C	d	SEQUENCE MULTIPLY OPTION: Enter the last n data entries, multiplied by d, into the current data string.

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.

FIXED-FIELD FIDO INPUT DETAILS

Cross sections may be input in a fixed-field FIDO format. But none of Blocks-I through VI input contain any fixed-field FIDO format. Details of the format are given below.

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. 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 subfields 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

   Character 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 consist of a subset of the operators used in the free-field input shown in Table 9.1 plus the special fixed field operators shown in Table 9.2. Only T, *, R, -, +, and Z may be used. In fixed-field FIDO usage of these operators, however, the following rules must be observed:

a. the "data numerator," if required, must be entered in the first subfield of a field;

b. the "data operator" must be entered in the second subfield of a field; and

c. the "data entry." if required, must be entered in the third subfield of a field.

Table 9.2 Special Fixed Field Data Operators

Data Numerator	Data Operator	Data Entry	Remarks
n	*		FLOATING ARRAY IDENTIFIER. The unique array number is n. Precedes the data in the array.
n	+	d	POSITIVE EXPONENTIATION. This will enter a single number with magnitude d x 10^{+n}, i.e., d is the mantissa and n is the tens exponent. If no decimal point is present in d, it will be assumed to be at the far right.
n	-	d	NEGATIVE EXPONENTIATION. This will enter a single number with magnitude d x 10^{-n}, i.e., d is the mantissa and n is the tens exponent. If no decimal point is present in d, it will be assumed to be at the far right.

REFERENCES

1. W. A. Rhoades and R. L. Childs, "An Updated Version of the DOT4 One- and Two-Dimensional Neutron/Photon Transport Code," Oak Ridge National Laboratory report ORNL-5851, (July 1982).

CROSS-SECTION LIBRARIES

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

10

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

In this chapter, we address the details of the many ways that multigroup cross sections can be supplied to the DANTSYS™ codes. We do NOT address the many issues involved in the processing of cross-section data into the multigroup format. Although the end user needs to be somewhat knowledgeable in that area, we feel that is outside the purview of this document.

The code cannot only read various cross-section library formats, but also has the capability to write several output formats. A few pages are devoted to these output formats and why one might want to use them.

Most of this chapter is concerned with the format and ordering of the data within each of several possible library files. This includes one section on how to order cross sections in a coupled library, such as a neutron-gamma library.

Then lastly, there is a small section about enforcing balance in the cross-section sets. This section touches briefly, but necessarily, on the production of the multigroup cross sections.

DANTSYS is a trademark of the Regents of the University of California, Los Alamos National Laboratory.

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.

ISOTXS and GRUPXS Standard Interface Files

Either of the standard interface files ISOTXS or GRUPXS 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. These are standard interface files as described by the Committee on Computer Code Coordination.[1,2] By default, the file wide vector chi (fission fractions) from these files will be used unless overridden by the zone dependent CHI in Block-V.

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. If LIB=ISOTXS, the cross sections, by default, must reside on a file named ISOTXS which must exist at the time of code execution. The user can override the name ISOTXS by setting LIBNAME=*filename* in the input, where *filename* is the name of the ISOTXS file. For LIB=GRUPXS, the same conditions apply.

Card-Image Libraries in Los Alamos, ANISN, or Free Field Format

The basic multigroup cross sections for isotopes can be provided in a card-image library whose form is referred to as Los Alamos, ANISN, or Free Field. This library form 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 such a card-image library by entering either LIB=ODNINP or LIB=XSLIB. If LIB= ODNINP, the library card-images are physically located within the input for the 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.

The user may also specify a card-image library by entering LIB=*filename* where *filename* is any name that the user chooses other than any of the following: ISOTXS, GRUPXS, MACRXS, MACBCD, BXSLIB, XSLIBB, MENDF, or MENDFG. The library card-images are then physically located on a file named *filename*.

Ordering of Cross Sections Within a Cross-Section Table

The Los Alamos, ANISN or FIDO card-image library form all assume that each cross-section table in the library contains an array of cross sections of NGROUP rows each containing IHM positions. The cross-section type for each group is determined by its position as shown in Table 10.1. Positions are specified relative to the positions of the total cross section σ_t (position IHT) and the within-group scattering cross section $\sigma_{g\rightarrow g}$, (position IHS). Note that the values of IHM, IHT, and IHS are user 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.

Table 10.1 Cross-Section Ordering In A Card-Image Library

Position in the Row	Cross-Section Type For Group g	Collective Designation
.	.	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]	$\sigma_s(g\rightarrow g)$	Self
IHS+1	$\sigma_s(g-1\rightarrow g)$	Downscatter Cross Sections
IHS+2	$\sigma_s(g-2\rightarrow g)$	
.	.	
.	.	
.	.	
IHM[a]	.	Last Position

a. IHM and IHS are user input in Block-III

Card-Image Data Formats

The cross-section data may be entered on the card-images in one of four data formats, the traditional Los Alamos format, another similar format with more accuracy, the fixed-field FIDO format, or the free-field 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 format.

For more accuracy a similar format (IFIDO= -1) is provided, for which the data are entered on the card-images in 4E18 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 "FIXED-FIELD FIDO INPUT DETAILS" on page 9-19. When this format is used, each cross-section table must be terminated with the "T" terminator as described there.

In the free-field format (IFIDO= 2), the data are entered on the card-images in free-field format as described in the "FREE FIELD INPUT DETAILS" on page 9-13. When this format is used, each cross-section table must be terminated with the "T" terminator described there.

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

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.

Anisotropic Scattering and the Ordering of Cross-Section Tables

In the code, it is assumed that the scattering transfer probability can be represented by the finite Legendre polynomial expansion of equation (14) on page 12-22, which, in multigroup notation, becomes

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

where $\mu_0 \equiv \underline{\Omega}' \cdot \underline{\Omega}$, 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 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 10.1. 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 high Legendre order cross-section tables, only the scattering cross sections are used. The first IHT rows for each group are ignored in the $P_L(L>0)$ 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.

Binary Form of Card-Image Libraries (the BXSLIB file)

The processing of large, card-image, ASCII 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, isotopic 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 library exists and is available to the code at the time of execution. To create the BXSLIB file, the user makes his initial execution with the card-image library (LIB= XSLIB or LIB= ODNINP) as previously described. Then, 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. For Los Alamos users, if LIB= MENDF, (see "The Los Alamos MENDF5 Cross-Section Library" on page 10-13) a BXSLIB file is always created and retained. 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 table labelled "Text Cross-Section Library Format" in the Block-III input description of any of the User's Guides. The information placed there will be that provided in the originating LIB= ODNINP or LIB= XSLIB run. Also included on the file is the input atomic weights, if they were input via card input in Block-IV. The file description for the BXSLIB binary file is provided in the chapter "FILE DESCRIPTIONS". See "BXSLIB" on page 15-38.

XSLIBB Card-Image Library File

By entering LIB=XSLIBB in the Block-III input, the user instructs the code to use the specialized isotopic cross-section file named XSLIBB. XSLIBB is an ASCII, card-image version of the BXSLIB file described previously. The principal advantage of an XSLIBB file is that it is an ASCII file and is thus both eye-readable and exportable. Since it can also contain other information such as NAMES, EDNAME, NTPI,

CHIVEC, VEL, ATWT, and EBOUND, it provides a very useful form of a cross-section library. Use of LIB=XSLIBB requires that the appropriate form of the card-image file exists with the name XSLIBB and is available to the code at the time of execution.

The creation of an XSLIBB file is controlled by the WRITMXS parameter in Block-III as described on page 10-19.

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 SNEXDT together with the standard interface files NDXSRF and ZNATDN[12] without referring to a basic library of multigroup isotope cross sections. These four files contain cross sections and other information pertaining to the materials created from the original isotopes. (A more detailed discussion of the MACRXS and SNXEDT file preparation process is provided in "INTERFACE FILES USED IN MIXING" on page 11-17 and file descriptions for MACRXS and SNXEDT are given in the chapter "FILE DESCRIPTIONS" starting on page 15-1.) 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 the code at the time of execution as follows:

a. MACRXS is required if the SOLVER module is to be executed, and

b. SNXEDT, NDXSRF, and ZNATDN are required if the EDIT module is to be executed.

The MACBCD Card-Image Cross-Section Library

By entering LIB=MACBCD in the Block-III input, the user instructs the code to use the specialized material cross-section file named MACBCD. MACBCD is an ASCII, card-image version of the MACRXS described in the preceding section. Use of LIB=MACBCD requires that the appropriate card-image file exists with the name MACBCD and that it is available to the code at the time of execution.

The creation of a MACBCD file is controlled by the WRITMXS parameter in Block-III as described later in the chapter.

The Los Alamos MENDF5 Cross-Section Library

At Los Alamos National Laboratory a multigroup, isotopic cross-section library named MENDF5 is maintained as a random access public file available to users on some of the Laboratory's mainframe computers. This file is derived from ENDF-V/B nuclear data evaluations. To use this binary library, or a library constructed in the same format as MENDF5, the user enters LIB=MENDF in the Block-III input. The code will seek a file

named MENDF in the user's local file space and, if such a file exists, will use it. If a file named MENDF does not exist in the local file space, the code will extract the MENDF5 public file and will use it instead.

The number of isotopes on the MENDF library must be entered using the NISO array of Block-I.

The appropriate fission fractions may be specified using the NTICHI input parameter in Block-III, or by using the CHIVEC input array of Block-III, or by using zone-dependent chi's in the Block-V input.

When using a MENDF file, isotopes are identified by a floating point number called a ZAID. The ZAID is of the form ZZAAA.NN where ZZ is the atomic number, AAA is the (three-digit) mass number, and NN is a two-digit number specifying a particular version of the cross sections for each given isotope. A listing of the current ZAIDs available on MENDF5 can be obtained from the Radiation Transport Group at Los Alamos National Laboratory.

The Los Alamos MENDF5G Gamma Cross-Section Library

Similar to the MENDF5 library at Los Alamos described in the preceding section, a multigroup, isotopic companion gamma-ray (photon) library is maintained. MENDF5G contains only neutron-induced photon-production and photon-interaction data. To access this library for gamma (photon)-only calculations, the user specifies LIB=MENDFG in Block-III of the input and the LNG (Last Neutron Group) parameter of Block-III must be set to zero.

The number of isotopes on the MENDFG library must be entered using the NISO array of Block-I.

Isotopes are referenced by their ZAIDs as described in the preceding section.

The XSLIBE and XSLIBF Material Cross-Section Libraries

The code can use card-image material cross-section libraries named XSLIBE and XSLIBF by setting LIB=XSLIBE or XSLIBF in Block-III of the input. These two files are ASCII, card-image files of the XSLIB type containing the material macroscopic cross sections taken from the MACRXS binary file. The format of these card-image files is the Los Alamos or ANISN format described on page 10-9 of this chapter. XSLIBE is formatted in the Los Alamos 6E12 card-image format while XSLIBF is formatted in the fixed-field FIDO format.

These files can be created by the code using the WRITMXS parameter in Block-III as described on page 10-19.

COUPLED NEUTRON-GAMMA CROSS SECTIONS

These codes 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.

As an example, a 42-group coupled set (NGROUP = 42) might have 30 neutron groups (LNG = 30) followed by 12 gamma groups. If there is no upscatter in the system, a coupled set can be provided in the form of a card-image library with 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., there is no neutron production via photoneutron, or gamma-n, reactions. Using the card-image form of a coupled library with cross sections ordered as shown in Table 10.2, "Arrangement of Data in a Coupled Neutron-Gamma Library Table," on page 10-16 (for no upscatter), the isotopic cross sections for each Legendre order of scatter carry data arranged as shown in Table 10.3. That table shows the contents of a card-image cross-section table for a 7-group coupled set (4 neutron, 3 gamma groups).

Table 10.2 Arrangement of Data in a Coupled Neutron-Gamma Library Table

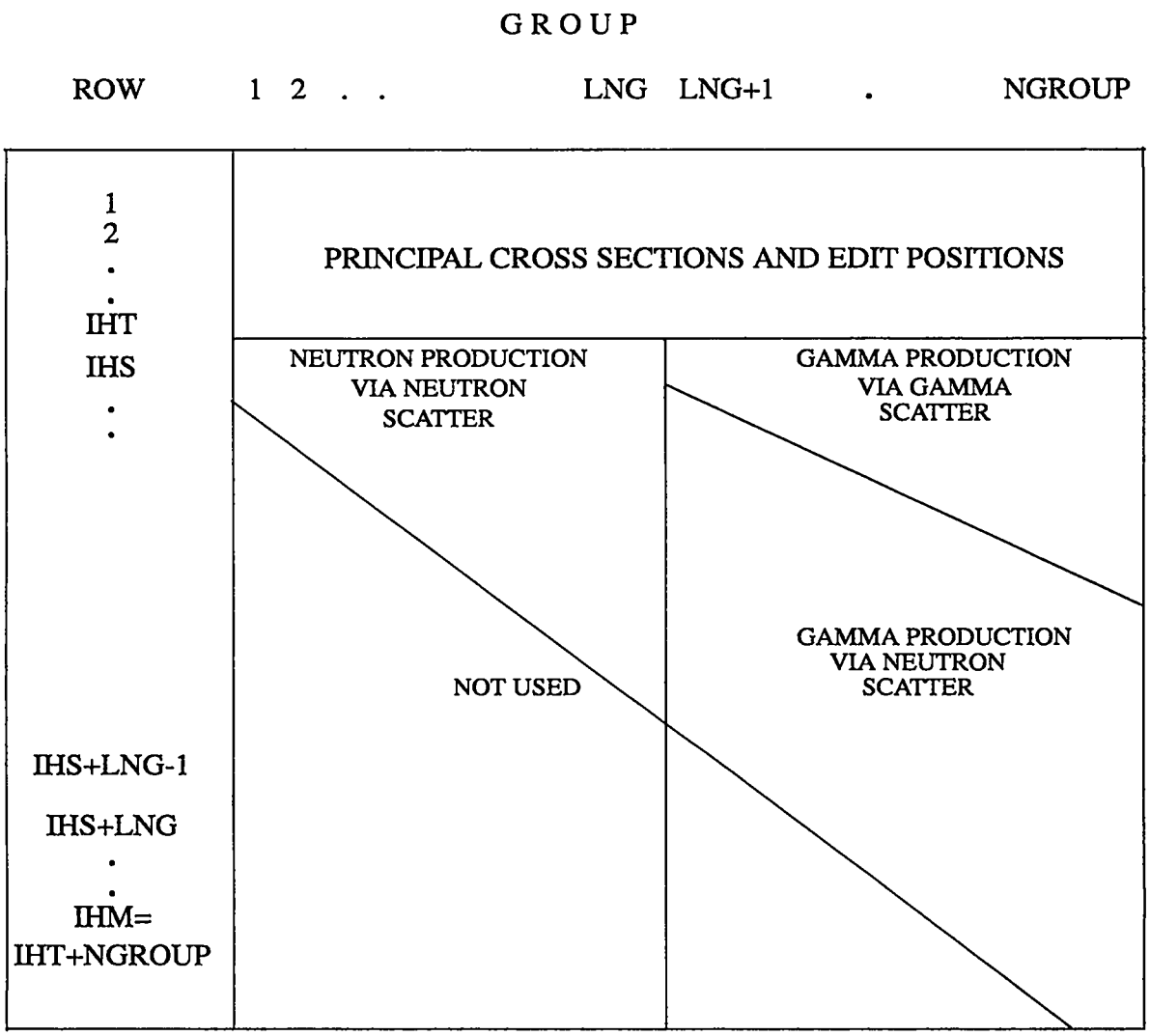


Table 10.3 Coupled[a] Cross-Section Table[b] Example

	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	$\sigma_\tau(1)$	$\sigma_\tau(2)$	$\sigma_t(3)$	$\sigma_\tau(4)$	$\sigma_\tau(5)$	$\sigma_\tau(6)$	$\sigma_\tau(7)$
4	$\sigma(1\rightarrow1)$	$\sigma(2\rightarrow2)$	$\sigma(3\rightarrow3)$	$\sigma(4\rightarrow4)$	$\sigma(5\rightarrow5)$	$\sigma(6\rightarrow6)$	$\sigma(7\rightarrow7)$
5	0	$\sigma(1\rightarrow2)$	$\sigma(2\rightarrow3)$	$\sigma(3\rightarrow4)$	$\sigma(4\rightarrow5)$	$\sigma(5\rightarrow6)$	$\sigma(6\rightarrow7)$
6	0	0	$\sigma(1\rightarrow3)$	$\sigma(2\rightarrow4)$	$\sigma(3\rightarrow5)$	$\sigma(4\rightarrow6)$	$\sigma(5\rightarrow7)$
7	0	0	0	$\sigma(1\rightarrow4)$	$\sigma(2\rightarrow5)$	$\sigma(3\rightarrow6)$	$\sigma(4\rightarrow7)$
8	0	0	0	0	$\sigma(1\rightarrow5)$	$\sigma(2\rightarrow6)$	$\sigma(3\rightarrow7)$
9	0	0	0	0	0	$\sigma(1\rightarrow6)$	$\sigma(2\rightarrow7)$
10	0	0	0	0	0	0	$\sigma(1\rightarrow7)$

a. 4 Neutron groups, 3 Gamma groups

b. Detailed specifications:
NGROUP = 7
LNG = 4
IHT = 3
IHS = 4
IHM = 10

CREATING FILES WITH DIFFERENT FORMATS

It is frequently desirable to produce card-image, ASCII forms of cross-section libraries that are eye-readable and exportable. The Input Module provides the user with the capability to produce several card-image library forms through the use of the WRITMXS parameter in Block-III of the input.

By setting WRITMXS=XSLIBB, the code is instructed to create the ASCII file XSLIBB described on page 10-12. It can be used if the original cross-section library (as specified in the LIB= instruction) is ODNINP, XSLIB, MENDF, MENDFG, or BXSLIB.

By entering WRITMXS=MACBCD, the code will create the ASCII file MACBCD described on page 10-13. The MACBCD file can be created no matter what the form of the original cross-section library.

By entering WRITMXS=XSLIBE, the code will create the ASCII file XSLIBE containing the material cross sections in Los Alamos 6E12 format as described on page 10-9 and page 10-14. Similarly, WRITMXS=XSLIBF instructs the code to create the ASCII file XSLIBF containing the material cross sections in fixed-field FIDO format as described on page 10-9 and page 10-14. Either file can be created irrespective of the form of the original cross-section library (as specified in the LIB= instruction).

BALANCING THE CROSS SECTIONS

The cross-section balancing option, triggered by the input variable, BALXS, should rarely be used. Proper use of BALXS requires a thorough understanding of the cross-section processing that went into the preparation of the multigroup cross-section library and why they are not balanced.

WARNING! Improper use of BALXS can INVALIDATE your flux solution.

In order to understand what BALXS does, we first discuss what the DANTSYS codes require of the cross-section set. When making the flux calculation for group g, it is important that the cross section for removal from the group be accurate. Removal from a group occurs either through absorption or through scattering to another group. The code implicitly gets the removal cross section, $\sigma_{r,g}$, from the total and self scattering cross sections:

$$\sigma_{r,g} = \sigma_{\text{total,g}} - \sigma_{g \to g} \tag{20}$$

Notice that nowhere is the absorption cross section, $\sigma_{\text{abs,g}}$, explicitly mentioned. It is NOT used directly or indirectly anywhere in the flux calculation. The flux calculation will give the correct results, given only correct values for $\sigma_{total,g}$ and $\sigma_{g \to g}$ (and, of course, the group to group transfer cross sections).

However, in order to provide the out-scattering rate for the balance table, the absorption cross section IS used. Recall that in the cross sections for a group, we have only the inscattering cross sections from other groups and not the outscattering cross section. One could access all the cross-section groups in order to sum up the outscatter from the source group, but conventionally, a short cut is used. The outscatter term is formed using an effective outscatter cross section, $\sigma_{out,g}$, calculated as shown in Eq. (21)

$$\sigma_{out,g} = \sigma_{r,g} - \sigma_{abs,g} \tag{21}$$

Now, consider the cross-section treatment for the (n,2n) reaction rate. Here we have, for each neutron lost to the source group, two neutrons appearing in other groups. Typically, this is handled by including the (n,2n) cross section in the total and the absorption positions of the source group, but then including a total of twice the (n,2n) cross section in the inscattering cross sections of the recipient groups. This procedure gives the correct removal cross section as represented by Eq. (20) and also accurately represents the inscattering source to the recipient groups. It also gives correctly the outscattering rate that shows in the balance table. However, the inscattering rate shown in the system balance table is formed from the inscatter cross sections and will include the (n,2n) production. The total inscatter rate (i.e. summed over the energy groups) will then be larger than the total outscatter rate by the amount of the production rate from the (n,2n) reaction.

Normally, the cross sections should balance, that is, the absorption reaction rate plus the total scattering reaction rate should equal the total reaction rate:

$$\sigma_{abs,g} + \sum_{g'} \sigma_{g \to g'} = \sigma_{total,g} \tag{22}$$

Again, recall that the transfer cross sections, $\sigma_{g \to g'}$, that we have in the library are normally the inscattering cross sections to group g' and not the outscattering cross sections from group g as required by Eq. (22). So, the cross sections will not be balanced according to Eq. (22) if there is any (n,2n). Typically, the remedy is to redefine the absorption cross section (recalling that this will not affect the flux calculation) to be:

$$\sigma'_{abs,g} = \sigma_{total,g} - \sum_{g'} \sigma_{g \to g'} \tag{23}$$

Where we are now using the only transfer cross sections we have, i.e., the inscattering cross sections, for $\sigma_{g \to g'}$. Note that $\sigma'_{abs,g}$ is then no longer the physical absorption cross section when we have any (n,2n). Also note that this new "$\sigma_{abs,g}$" will cause the total outscattering in the system balance table to be equal to the total inscattering even in the presence of (n,2n), and hence, you will get the true system balance there.

The bottom line is that if your cross-section processor supplies a balanced set with the proper total cross section, the proper scattering cross sections in the transfer matrix, and an absorption cross section satisfying Eq. (23), you should get the correct flux calculation and the correct balance calculated in the balance table even in the presence of (n,2n).

However, let us suppose that the cross-section purveyor supplies the true absorption cross section rather than that shown in Eq. (23) and that there is significant (n,2n) in the material. The cross-section set will not then be balanced. According to the earlier discussion, you will still get the correct flux solution, but the calculated system balance will be artificially off due solely to this peculiarity of the cross sections. The balance then is not a good measure of how well the iteration procedure went in determining the flux. You can use the input option, BALXS = -1, to replace the supplied $\sigma_{abs,g}$ with that calculated from Eq. (23), again recalling that changes in the absorption cross section will not affect the flux calculation, but will remove the artificial part of the imbalance.

If you use the other option, BALXS = +1, the self scattering cross section in Eq. (23) will be adjusted to balance the cross sections. But as this WILL change the flux solution because you are then changing the removal cross section, you must know that you are correcting an error in the cross-section processing when you use this option. If you force cross-section balancing this way when it is inappropriate, you will get an INCORRECT flux solution.

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).

MATERIAL MIXING TUTORIAL

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

11

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF TABLES

REVIEW OF TERMINOLOGY

An understanding of the general procedure for mixing nuclides in these codes requires an understanding of the terminology used with the code. A review of five basic terms is provided below.

1. Fine Mesh - the Fine Mesh is the spatial solution mesh for the problem.

2. Coarse Mesh - the Coarse Mesh is a spatial superset of the Fine Mesh. Each Coarse Mesh contains one or more Fine Mesh intervals. Fine mesh widths are all the same within a given coarse mesh interval. No material discontinuities may occur within a coarse mesh interval.

3. Zone - the Zone is a spatial superset of the Coarse Mesh intervals. Each Zone contains one or more Coarse Mesh intervals. A Zone is characterized by a single set of macroscopic multigroup cross sections so that all fine mesh intervals within a zone have the same cross sections.

4. Material - a Material is composed of isotopes (or nuclides or mixes) whose "microscopic" cross sections exist on the cross-section library file to be used. The specification of materials requires knowing which isotopes and how much of each isotope goes into the Material. This information is provided the code through the MATLS= array in Block-IV of the input.

5. Material Assignments to Zones - the material composition of a Zone is made by assigning one or more Materials in specified amounts to each Zone through the ASSIGN= input array in Block-IV of the input. This assignment thus links a spatial portion of the physical problem model (the zone) to the macroscopic cross sections that are to be used in that spatially-defined zone.

MIXING MATERIALS FROM "ISOTOPES"

If "isotopic" cross sections are provided on a cross-section library, it is necessary to mix these isotopes, or nuclides, 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, the specifications of which are described in any of the user's guides chapters for ONEDANT, TWODANT, or THREEDANT, or by (ii) the standard interface files NDXSRF and ZNATDN.[1] If the NDXSRF and ZNATDN files are used, the term "zone" in the file descriptions of Ref. 1 and Ref. 2 must be replaced with the word "material" to be consistent with our terminology.

In this section we will provide a basic description of how to mix isotopes (or nuclides) to form materials using the MATLS array specification in Block-IV of the input. For now we will assume that it is desired to create materials by specifying the isotopes and their atom densities in each material. Later in this chapter will be described some alternatives to using atom densities.

Let us consider an example of a common way of specifying materials. Suppose we have a multigroup library of cross sections (in barns) for several nuclides including those whose identifiers are U235, U238, PU239, PU240, IRON, CHROME, NICKEL, OXYGEN, and SODIUM. It is desired to mix these nuclides into the following materials:

1. Stainless Steel (SS) at 8 gm/cm^3
   [74% Iron, 18% Chromium, and 8% Nickel (percents are weight percents)],

2. Uranium Dioxide (UO2) at 10 gm/cm^3
   [0.3% U235 and 99.7% U238 (percents are atomic percents)],

3. Plutonium Dioxide (PUO2) at 10 gm/cm^3
   [80% PU239 and 20% PU240 (percents are atomic percents)],

4. Sodium (NA) at 0.8 gm/cm^3.

Translating the above information into atom densities (atoms/barn-cm) gives the values shown in Table 11.1.

Table 11.1 Example Specification of Materials

Material Identifier	Isotope Identifier	Atom Density
SS	CHROME	1.6676E-2
	NICKEL	6.566E-3
	IRON	6.3832E-2
UO2	U235	6.7E-5
	U238	2.223E-2
	OXYGEN	4.460E-2
PUO2	PU239	1.7761E-2
	PU240	4.440E-3
	OXYGEN	4.4402E-2
NA	SODIUM	2.096E-2

To input this information for any DANTSYS solver, we enter in Block-IV the following:

```
MATLS= SS,    CHROME  1.6676-2, NICKEL 6.566-3,   IRON 6.3832-2 ;
       UO2,   U235    6.7-5,    U238   2.2234-2, OXYGEN 4.4602-2 ;
       PUO2,  PU239   1.7761-2, PU240  4.44-3,    OXYGEN 4.4402-2 ;
       NA,    SODIUM  2.096-2 ;
```

Note the use of semicolons (;) to delimit the different materials indicating a separate string for each material (see the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1). Also note that the use of commas is optional. Blanks also serve as delimiters.

ASSIGNING 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 (U8O2), U-235 Oxide (U5O2), and PU-239 Oxide (PU9O2). 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:

Table 11.2 Composition of Zones

Zone	Material	Material Volume Fraction
CORE	SS	0.25
	NA	0.40
	U8O2	0.20
	PU9O2	0.15
BLKT	SS	0.25
	NA	0.40
	U8O2	0.349
	U5O2	0.001
REFL	SS	0.030
	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   U8O2 0.20   PU9O2 0.15 ;
        BLKT  SS 0.25   NA 0.40   U8O2 0.349  U5O2 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 the chapter "FILE DESCRIPTIONS" starting on page 15-1.

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 at the time of code execution.

ALTERNATIVE FORMS OF MIXING

This section details the use of the MATSPEC and ATWT arrays.

Historically, DANTSYS has been oriented toward specifying materials by requiring input that gives the atom densities of isotopes comprising the material. This is typical of reactor-oriented computer codes. There is a nuclear analysis community, however, that specifies mixes by identifying materials by the density of the material together with the isotopes and their atomic or weight fractions in that material.

The DANTSYS codes will accept the latter type of material specification in addition to the traditional atom density type of specification. A description follows.

Using Atomic Fractions or Weight Fractions (MATSPEC)

In Block-IV of the DANTSYS input (the mixing specifications) is the parameter MATSPEC. There are three allowable input values for MATSPEC:

a. MATSPEC=ATDENS tells the code you are using the traditional atom density style of input for mixing specifications as described on page 11-9. This value is the DEFAULT.

b. MATSPEC=ATFRAC tells the code you are using the type of mixing that specifies the density (gm/cc) of each material together with the isotopes (nuclides) and their atomic fractions in each material.

c. MATSPEC=WTFRAC tells the code you are using the type of mixing that specifies the density (gm/cc) of each material together with the isotopes (nuclides) and their weight fractions in each material.

The MATSPEC parameter can be entered as a vector parameter with up to MT entries so that different materials can be specified with different types of mixing specifications. (Recall that MT is the number of materials to be created, as specified in Block-I of the input.) If the number of entries is less than MT, the last entry will be applied to all remaining and unspecified entries.

Following is a description of how to input information in each of the above styles.

Note: In the following, mat_m denotes the name (or identifier) of material m. It is usually a character name of the user's choice, e.g., fuel, Tu, steel, etc., for ease of reading.

$iso_{i,m}$ denotes the name (or identifier) of an isotope that exists in material m. This must be one of the names from the cross-section library. It is the i-th isotope in the specific list for material m and may or may not be the i-th isotope

on the cross-section library. $iso_{i,m}$ is usually the ZAID identifier* if one is using a MENDF cross-section library.

$af_{i,m}$ denotes the atomic fraction of isotope i that exists in material m.

$wf_{i,m}$ denotes the weight fraction of isotope i that exists in material m.

$zonid_j$ denotes the name (or identifier) of zone j. It is usually a hollerith name, of the user's choice, e.g. core, shell, driver, etc., for ease of reading and identifying.

ρ_m denotes the density [gm/cc] of material m as it exists in a zone.

- MATSPEC= ATDENS

If one makes the entry MATSPEC=ATDENS or one omits the MATSPEC= entry, the mixing is done with atom densities as described earlier in this chapter.

- MATSPEC= ATFRAC

(Step 1) One defines materials, each with a density of 1 gm/cc, by specifying (using the MATLS= input array of Block-IV) the name (or identifier) of the material followed by the isotope names (or identifiers) paired with the atomic fractions of those isotopes which define the material. This is done by inputting

MATLS= mat_1 $iso_{1,1}$ $af_{1,1}$ $iso_{2,1}$ $af_{2,1}$ ... ; mat_2 $iso_{1,2}$ $af_{1,2}$ $iso_{2,2}$, $af_{2,2}$ ... ; etc.

NOTE: Each material's specification must be separated from the next by a semicolon!

(Step 2) Then one specifies how the above materials will be mixed and at what density into each zone. This is done by inputting

ASSIGN= $zonid_1$ mat_i ρ_i ... ; $zonid_2$ mat_k ρ_k mat_l ρ_l ... ; etc.

NOTE: Each zone's specification must be separated from the next by a semicolon!

* The user may get a list of the available ZAID's by making a preliminary run that only includes the Block-I and Block-III input. A listing of the ZAID's will be found in the printed output from that run.

• MATSPEC= WTFRAC

(Step 1) One defines materials, each with a density of 1 gm/cc, by specifying (using the MATLS= input array of Block-IV) the name (or identifier) of the material followed by the isotope names (or identifiers) paired with the weight fractions of those isotopes which define the material. This is done by inputting

MATLS= mat_1 $iso_{1,1}$ $wf_{1,1}$ $iso_{2,1}$ $wf_{2,1}$ ... ; mat_2 $iso_{1,2}$ $wf_{1,2}$ $iso_{2,2}$ $wf_{2,2}$ ... ; etc.

NOTE: Each material's specification must be separated from the next by a semicolon!

(Step 2) Then one specifies how the above materials will be mixed and at what density into each zone. This is done by inputting

ASSIGN= $zonid_1$ $mati$ ρ_i ... ; $zonid_2$ mat_k ρ_k mat_l ρ_l ... ; etc.

NOTE: Each zone's specification must be separated from the next by a semicolon!

To illustrate the use of these arrays, we repeat the example of page 11-9 with the following:

```
MATSPEC= WTFRAC  ATFRAC

MATLS=SS,  IRON     0.74,     CHROME0.18,      NICKEL0.08;
      UO2, U235     0.003     U238     0.997   OXYGEN2.0;
           PUO2,PU2390.8       PU240    0.2     OXYGEN2.0;
           NA, SODIUM1.0;

ASSIGN=STEEL SS     8.0;
           UFUELUO210.0;
           PFUEL PUO210.0;
           COOLNA0.0;
```

Note that the density of the materials are put in the ASSIGN array whereas the respective fractions defined by the MATSPEC array are in the MATLS card. Note also that we rely on the word, ATFRAC, being repeated to fill the third and fourth positions of the MATSPEC array.

Providing Atomic Weights to the Code (the ATWT Array)

When using MATSPEC=ATFRAC or MATSPEC=WTFRAC, the code must have the atomic weights of the isotopes.

If the cross-section library is not a MENDF, MENDFG, or a BXSLIB file created with atomic weights on it, the atomic weights must be supplied in the Block-IV input. This is done as follows:

In Block-IV enter

$$\text{ATWT}= iso_1 \ atwt_1 \ \ iso_2 \ atwt_2 \ \ldots \ iso_n \ atwt_n$$

where $n \leq$ NISO, iso_i is the isotope name (identifier) for an isotope on the cross-section library, and $atwt_i$ is its atomic weight. It is the i-th isotope in this specific list of atomic weights and may or may not be the i-th isotope on the cross-section library.

If using LIB=MENDF for neutrons or LIB=MENDFG for gammas in Block-III of the input, the atomic weights are automatically provided and nothing more need be done by the user.

NOTE: Whenever the atomic weights are provided, either in the card-image input or from a MENDF or MENDFG file, the BXSLIB file that the code creates will contain the atomic weight data automatically. By saving this BXSLIB file (see the SAVBXS parameter in Block-III) and using it as the cross-section library file in subsequent runs, there will be no need to re-enter the atomic weights in the card-image input.

INTERFACE FILES USED IN MIXING

1. Material Mixing and the Creation of Interface Files.

In the material mixing operation in the Input Module of DANTSYS, 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 isotope cross-section library is "forgotten" once these four files are created.

The MACRXS code-dependent binary interface file is described in the chapter "FILE DESCRIPTIONS" starting on page 15-1 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, for example, 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 in Block-IV, a single set of materials, that is, a single MACRXS file can be used for calculating numerous different problems in which the problem zone compositions consist of different proportions of materials. See "ASSIGNING MATERIALS TO ZONES" on page 11-11. The manner in which the code is instructed to use an existing MACRXS file is described below on page 11-18.

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 the chapter "RUNNING THE EDIT MODULE" starting on page 8-1. The SNXEDT file description is given in the chapter "FILE DESCRIPTIONS" on page 15-78.

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 the edit section of any of the User's Guides. It is again noted that in using the NDXSRF and ZNATDN files, the term "zone" in the file descriptions in Ref. 1 and Ref. 2 must be replaced with the word "material" to be consistent with our 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 in Block-V. See "ASSIGNING MATERIALS TO ZONES" on page 11-11.

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 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 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-III 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 at the time of execution. Note: only the MACRXS, NDXSRF, and ZNATDN files are needed for execution of the Solver Module, and only the SNXEDT, NDXSRF, and ZNATDN files are needed for execution of the Edit Module.

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).

ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT – Methods Manual

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

12

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This chapter provides the development of the multigroup, discrete-ordinates, form of the time-independent Boltzmann transport equation.[1,2] This development is followed by a brief description of the iterative procedure used to solve the transport equation employing the diffusion synthetic acceleration (DSA) scheme to accelerate the iterations[3]. The specifics of the discrete ordinates method is provided for the ONEDANT, TWODANT, TWODANT/GQ, and THREEDANT solver modules. We also include some of the details on spatial discretization as is appropriate for each geometry.

The First Order Form of the Boltzmann Transport Equation

The time-independent inhomogeneous Boltzmann transport equation may be written as:

$$\begin{aligned}\nabla \cdot \underline{\Omega}\Psi(r, E, \underline{\Omega}) + \sigma(r, E)\,\psi(r, E, \underline{\Omega}) &= \iint dE' d\Omega' \sigma_s(r, E' \rightarrow E, \underline{\Omega} \cdot \underline{\Omega}')\,\psi(r, E', \underline{\Omega}') \\ &+ \frac{1}{k_{eff}} \iint dE' d\Omega' \chi(r, E' \rightarrow E)\,\nu\sigma_f(r, E')\,\psi(r, E', \underline{\Omega}') \\ &+ Q(r, E, \underline{\Omega}),\end{aligned} \tag{1}$$

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. Assuming isotropic media, the macroscopic scattering transfer cross section, 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 $\left(\frac{1}{4\pi}\right)$ per fission is ν, and the fraction of these particles appearing in energy dE about E from fissions in dE′ about E′ is $\chi(r, E' \rightarrow E)$. If Q is not zero in Eq. (1), then k_{eff} is set to 1; if Q is zero, then the problem is an eigenvalue problem and $1/k_{eff}$ is the eigenvalue.

The remainder of this chapter is devoted to the consideration of the discretization of this equation as employed in each of the solver modules. The energy and angular variable discretization is common to all the modules and hence is developed first. The spatial discretization is explained separately for each of the modules concerned.

MULTIGROUP, DISCRETE ORDINATES TRANSPORT THEORY

We begin the discretization of the neutral particle, Boltzmann transport equation by considering the phase space variables of energy and angle. This is the multigroup in energy and discrete ordinates in angle approximation which is common to all the solver modules in the DANTSYS code system.

Multigroup Approximation

In the energy variable, the energy domain is partitioned into G intervals of width $\Delta E_g = E_{g-1/2} - E_{g+1/2}$. By convention, the highest energy is at $E_{1/2}$ and hence the index g increases as energy decreases (since the normal transport of particles in energy is from high to low energy as they collide with nuclei in the medium). Thus if we integrate Eq. (1) over each energy interval, we obtain the following discretized equation:

$$\underline{\Omega} \bullet \nabla \psi_g (\underline{r}, \underline{\Omega}) + \sigma_{t,g} (\underline{r}) \psi_g (\underline{r}, \underline{\Omega}) = \frac{\chi_g (\underline{r})}{k_{eff}} \Phi (\underline{r}) + Q_g (\underline{r}, \underline{\Omega}) + \tag{2}$$

$$\sum_{g'=1}^{G} \int \sigma_{s, g' \rightarrow g} (\underline{r}, \mu_0) \psi_{g'} (\underline{r}, \underline{\Omega}') \, d\underline{\Omega}'$$

$$g = 1, \ldots, G$$

where, $\mu_0 = \underline{\Omega} \bullet \underline{\Omega}'$,

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

$$\sigma_{t,g} (\underline{r}) = \left(\int_{\Delta E_g} \sigma_t (\underline{r}, E) \psi (\underline{r}, \underline{\Omega}, E) \, dE \right) / \psi_g (\underline{r}, \underline{\Omega}),$$

$$\sigma_{s, g' \rightarrow g} (\underline{r}, \mu_0) = \left(\int_{\Delta E_g} \int_{\Delta E_{g'}} \sigma (\underline{r}, E' \rightarrow E, \mu_0) \psi (\underline{r}, \underline{\Omega}, E') \, dE dE' \right) / \psi_{g'} (\underline{r}, \underline{\Omega}),$$

$$Q_g(\underline{r}, \underline{\Omega}) = \int_{\Delta E_g} Q(\underline{r}, \underline{\Omega}, E)\, dE,$$

$$\chi_g(\underline{r}) = \int_{\Delta E_g} \chi(\underline{r}, E)\, dE,$$

$$\Phi(\underline{r}) = \sum_{g=1}^{G} \left(\int_{\Delta E_g} \nu\sigma_f(\underline{r}, E)\, \phi(\underline{r}, E)\, dE \right) / (\phi_g(r)),$$

$$\phi_g(\underline{r}) = \int \psi_g(\underline{r}, \underline{\Omega})\, d\underline{\Omega}.$$

Note that the definition of the multigroup cross sections is a formal one since one would need to know the solution of the transport problem in order to evaluate them. In practice, the multigroup cross sections are supplied from some cross-section processing code which predefines the energy intervals and the weighting functions. We also note that much of the physics of the problem is contained in the cross-section set and hence this aspect of the solution process of the transport problem should not be minimized. That is, a careful selection of the cross-section set should be made in order to ensure a physically accurate solution. For few groups, this accuracy is heavily dependent upon the weighting function; as the number of groups increases, this becomes less so assuming a proper treatment of the resonance region.

Discrete Ordinates Approximation

The next discretization involves the angular variable. In the method of discrete ordinates, the angle is discretized over the unit sphere in a prescribed manner. The choice of the discretization seeks to satisfy the following conditions:

(1) physical symmetries are preserved upon discretization,

(2) spherical harmonics moments are well approximated in order to provide an accurate representation of the source terms, and

(3) derivatives with respect to the angle coordinates that come from the streaming operator are simply approximated.

Since in multidimensional cases not all of the above conditions can be simultaneously satisfied, compromises are made in defining discrete ordinates sets. To understand these points more completely see Ref. 4. Basically each discrete ordinate is depicted as an angular direction vector, Ω_m, with an associated area on the unit sphere, w_m, where m=1,... M, M being the number of discrete ordinates. This M is derived form the S_n order specified in Block-I and depends upon the set arrangement chosen and the number of spatial dimensions. All of the discrete ordinates sets used in the solver modules satisfy certain, fundamental conditions including $\sum_{m=1}^{M} w_m = 4\pi$ *for conservation. We also impose the requirements that:

$$\sum_{m=1}^{M} w_m \underline{\Omega}_m = 0 \text{ for symmetry, and}$$

$$\sum_{m=1}^{M} w_m \mu_m^2 = \sum_{m=1}^{M} w_m \eta_m^2 = \sum_{m=1}^{M} w_m \xi_m^2 = \frac{1}{3} \text{ for the diffusion limit.}$$

where the components of $\underline{\Omega}_m$ are: $\underline{\Omega}_m = \mu_m \underline{i} + \eta_m \underline{j} + \xi_m \underline{k}$.

To obtain the discrete ordinates balance equation, we integrate Eq (2) over w_m (in this case we assume isotropic scattering for simplicity):

$$[\underline{\Omega} \bullet \nabla \psi_g(\underline{r}, \underline{\Omega})]_m + \sigma_{t,g}(\underline{r}) \psi_{g,m}(\underline{r}) = \frac{\chi_g(\underline{r})}{k_{eff}} \Phi(\underline{r}) + Q_{g,m}(\underline{r}) + \sum_{g'=1}^{G} \sigma_{s0,g' \to g}(\underline{r}) \phi_{g'}(\underline{r}) \quad (3)$$

where,

$$\psi_{g,m}(\underline{r}) = \int_{w_m} \psi(\underline{r}, \underline{\Omega}) \, d\underline{\Omega},$$

$$\phi_g(\underline{r}) = \sum_{m=1}^{M} w_m \psi_{g,m}(\underline{r}),$$

$$Q_{g,m}(\underline{r}) = \sum_{l=0}^{L} \sum_{q=-l}^{l} (2l+1) Y_l^q(\underline{\Omega}_m) Q_{g,l}^q(\underline{r}),$$

$$Q_{g,l}^q(\underline{r}) = \int Y_l^{*q}(\underline{\Omega}) Q_g(\underline{r}, \underline{\Omega}) \, d\underline{\Omega},$$

and the $Y_l^q(\underline{\Omega})$ are the spherical harmonic functions.

Note that the streaming operator is only discretized symbolically; the specific form of this operator depends upon the geometric symmetry chosen and to some extent the spatial discretization method. We also note that we have expanded the source in spherical harmonics, the form in which the code expects the angular dependence of the source to be represented if it is a volume source. Discussion of the representation of an anisotropic

*In this code we renormalize the weights such that $\sum_{m=1}^{M} w_m = 1$.

scattering source is presented in the section "Spherical Harmonics Expansion of the Scattering Source" on page 12-22 below.

Iteration Procedure and Diffusion Synthetic Acceleration

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 DANTSYS Solver Modules (except TWOHEX) employ the diffusion synthetic acceleration (DSA) method developed by Alcouffe,[3] 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

$$\underline{\Omega} \cdot \nabla \tilde{\psi}_g^l(r, \underline{\Omega}) + \sigma_g(r)\, \tilde{\psi}_g^l(r, \underline{\Omega}) = \sigma_{s, g \to g}(r)\, \phi_g^{l-1}(r) + QQ_g(r). \tag{4}$$

In Eq. (4), $\tilde{\psi}_g^l(r, \underline{\Omega})$ is the angular flux for group g at the l^{th} inner iteration using a scalar flux $\phi_g^{l-1}(r)$ assumed known from the previous 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^l needed for the next iteration. In 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

$$-\nabla \cdot D_g(r)\, \nabla \phi_g^l(r) + \sigma_{R,g}(r)\, \phi_g^l(r) = QQ_g(r) - R_g^l(r), \tag{5}$$

where

$$D_g(r) = \frac{1}{3\sigma_{tr,g}}(r), \qquad \sigma_{R,g}(r) = \sigma_g(r) - \sigma_{s, g \to g}(r)$$

and the correction term is

$$R_g^l(r) = \nabla \cdot \tilde{J}_g^l(r) + \nabla \cdot D_g(r)\, \nabla \tilde{\phi}_g^l(r). \tag{6}$$

In Eq. (6),

$$\tilde{\phi}_g^l(r) = \int d\Omega \tilde{\psi}_g^l(r, \underline{\Omega}), \qquad \tilde{J}_g^l(r) = \int d\Omega \underline{\Omega} \tilde{\psi}_g^l(r, \underline{\Omega}). \tag{7}$$

Note that a tilde is used to indicate quantities calculated using the transport angular flux, $\tilde{\psi}_g^l$, 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 ϕ_g^{l-1}, known from the previous iteration, Eq. (4) is solved for $\tilde{\psi}_g^l$. This involves one sweep through the space-angle mesh. The correction term, R_g^l, is then calculated using Eqs. (6) and (7) and, in turn, used in Eq. (5) 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. (5) with R_g set to zero for l=0.

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. (6) into Eq. (5) yields

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

which is the converged transport balance equation obtained also by integrating Eq. (4) over all $\underline{\Omega}$.

After performing the inner iterations for each group using Eqs. (4) through (7) to obtain the group converged correction terms $R_g^k(r)$, a second level of iteration is needed to update the multigroup sources. A multigroup, corrected diffusion equation is constructed for this purpose which has the following form:

$$\begin{aligned} -\nabla \cdot D_g(r)\nabla\phi_g^{k+1}(r) + \sigma_{R,g}(r)\phi_g^{k+1}(r) &= Q_g(r) - R_g^k(r) \\ + \chi_g \sum_{g'=1}^{G} \nu\sigma_{f,g'}(r)\phi_{g'}^{k+1}(r) &+ \sum_{g' \neq g} \sigma_{s,g' \to g}(r)\phi_{g'}^{k+1}(r) \end{aligned} \tag{8}$$

Equation (8) is necessary if there are fissions or upscattering processes in the problem. The solution of this multigroup, diffusion equation in itself involves an iteration procedure which we term sub-outer iteration. In using this second level of iteration, we are effectively replacing transport outer iterations by the diffusion sub-outers.

The source correction scheme outlined above using the diffusion synthetic method is an effective scheme for inhomogeneous source problems. For eigenvalue problems, Eq. (8) 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 $\underline{\underline{D}}_g(r)$ as

$$\underline{\underline{D}}_g(r) = \frac{-\tilde{J}_g(r)}{\nabla\tilde{\phi}_g(r)}, \tag{9}$$

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

$$-\nabla \cdot \underline{\underline{D}}_g^{l-1}(r) \cdot \nabla\phi_g^l(r) + \sigma_{R,g}(r)\,\phi_g^l(r) = QQ_g(r), \tag{10}$$

and the multigroup (outer iteration) diffusion equation becomes

$$\begin{aligned} -\nabla \cdot \underline{\underline{D}}_g^k(r) \cdot \nabla\phi_g^{k+1}(r) + \sigma_{R,g}(r)\,\phi_g^{k+1}(r) &= \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu\sigma_{f,g'}(r)\,\phi_{g'}^{k+1}(r) \\ &+ \sum_{g' \neq g} \sigma_{s,g' \to g}(r)\,\phi_{g'}^{k+1}(r), \end{aligned} \tag{11}$$

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, TWODANT, TWODANT/GQ, and THREEDANT, the diffusion coefficient correction scheme is used for inhomogeneous source problems in which fission and/or upscatter is present while the source correction scheme is 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. (9)). If this occurs, Eq. (10) cannot be solved using current techniques. To overcome this difficulty, the removal correction scheme is employed. A corrected removal cross section is defined as

$$\tilde{\sigma}_{R,g}^l(r) \equiv \sigma_{R,g}(r) + \frac{R_g^l(r)}{\tilde{\phi}_g^l(r)}, \tag{12}$$

where $R_g^l(r)$ is defined by Eq. (6). With this modification, Eq. (5) becomes

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

and Eq. (8) becomes

$$-\nabla \cdot D_g(r)\nabla\phi_g^{k+1}(r) + \tilde{\sigma}_{R,g}(r)\phi_g^{k+1}(r)$$
$$= \frac{\chi_g}{k_{eff}} \sum_{g'=1}^{G} \nu\sigma_{f,g'}(r)\phi_{g'}^{k+1} + \sum_{g' \neq g} \sigma_{s,g' \to g}(r)\phi_{g'}^{k+1}(r). \quad (13)$$

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.

ONEDANT METHODS

This section is devoted to the description of the angular and spatial discretization forms that are specific to the one-dimensional geometries treated in ONEDANT. The one-dimensional symmetries offer great simplification of the transport process and allow rapid transport calculations for physical systems that can be reasonably represented as one dimensional.

Geometrical Symmetries Treated in ONEDANT

The symmetries treated in ONEDANT are: (1) slab, (2) infinite cylinder, (3) sphere, and (4) two-angle slab. The two-angle slab case treats slab geometries where the angular dependence of an incident source (boundary source) is possibly outside the plane normal to the surface of the slab.

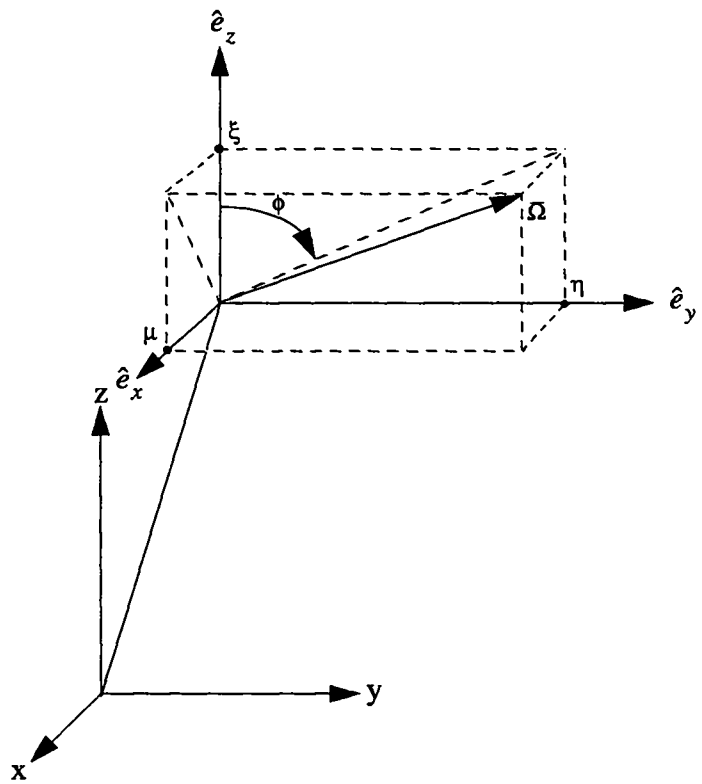


Figure 12.1 Coordinates in plane geometry

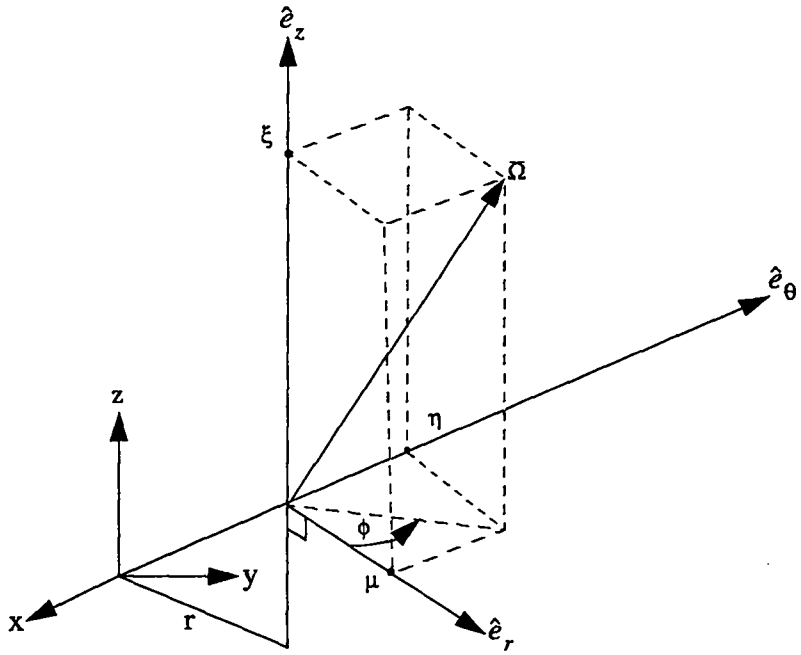


Figure 12.2 Coordinates in cylindrical geometry

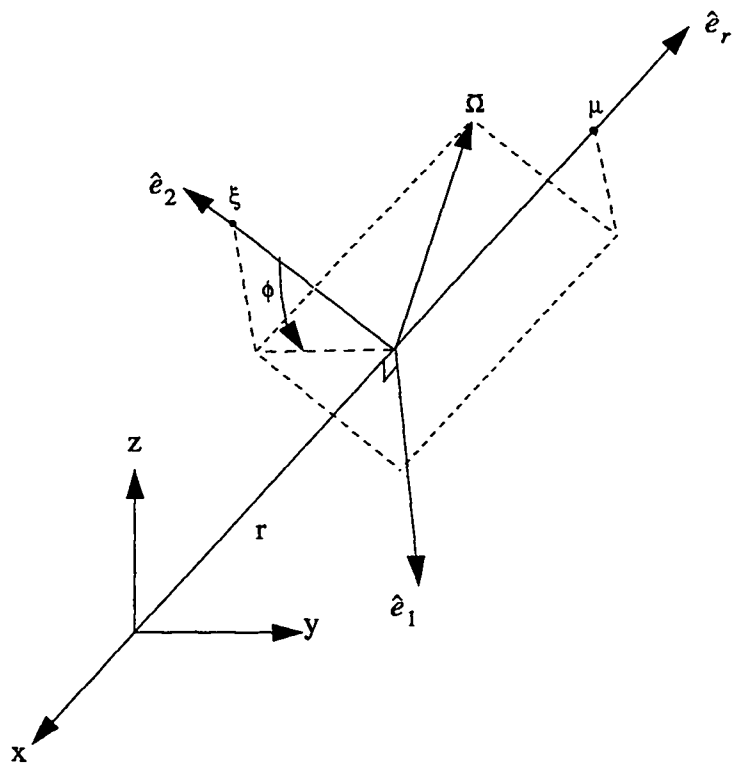


Figure 12.3 Coordinates in spherical geometry

Particular Forms of the Divergence Operator

The divergence operator in conservative form, $\nabla \cdot \underline{\Omega}\psi$ or $(\underline{\Omega} \cdot \nabla\psi)$ for the geometries treated by ONEDANT is given in Table 12.1 in terms of the coordinate systems shown in Figure 12.1 through Figure 12.3.

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 $\mu - \xi$, (or $\phi = 0° - 180°$) 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 interval (-1, +1).

Table 12.1 Forms of the Divergence Operator

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

Spherical Harmonics Expansion of the Scattering Source

The scattering transfer cross section in Eq. (1) is represented by a finite Legendre polynomial expansion of order ISCT

$$\sigma_s(r, E' \to E, \underline{\Omega}' \cdot \underline{\Omega}) = \sum_{L=0}^{ISCT} \left(\frac{2L+1}{4\pi}\right) \sigma_s^L(r, E' \to E) P_L(\underline{\Omega}' \cdot \underline{\Omega}) \tag{14}$$

If this expansion is inserted into Eq. (1) and the addition theorem for spherical harmonics used to expand $P_n(\underline{\Omega}' \cdot \underline{\Omega})$, the scattering source becomes

$$\iint dE' d\Omega' \sigma_s(r, E' \to E, \underline{\Omega} \cdot \underline{\Omega}') \psi(r, E', \underline{\Omega}') \equiv SS$$

$$SS = \int_E dE' \sum_{L=0}^{ISCT} \left(\frac{2L+1}{4\pi}\right) \sigma_S^L(r, E' \to E) \left\{ P_L(\mu) \int_{-1}^{1} d\mu' \int_0^{2\pi} d\phi' P_L(\mu') \psi(r, E', \mu', \phi') \right.$$
$$\left. + 2 \sum_{K=1}^{L} \frac{(L-K)!}{(L+K)!} P_L^K(\mu) \int_{-1}^{1} d\mu' \int_0^{2\pi} d\phi' P_L^K(\mu') \cos K(\phi - \phi') \psi(r, E', \mu', \phi') \right\} \tag{15}$$

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. (15) as

$$SS = \int_E dE' \sum_{L=0}^{ISCT} (2L+1) \sigma_S^L(r, E' \to E) \left\{ P_L(\mu) \phi_L(r, E') \right.$$
$$\left. + \sum_{K=1}^{L} \sqrt{\frac{2(L-K)!}{(L+K)!}} \left[\phi_{C,L}^K(r, E') P_L^K(\mu) \cos K\phi + \phi_{S,L}^K(r, E') P_L^K(\mu) \sin K\phi \right] \right\} \tag{16}$$

where we have defined the moments of the angular flux as

$$\phi_L(r, E') \equiv \frac{1}{4\pi} \int_{-1}^{1} d\mu' \int_0^{2\pi} d\phi' P_L(\mu') \psi(r, E', \mu', \phi'), \tag{17a}$$

$$\phi_{C,L}^K(r, E') \equiv \frac{1}{4\pi} \int_{-1}^{1} d\mu' \int_0^{2\pi} d\phi' \psi(r, E', \mu', \phi') \sqrt{\frac{2(L-K)!}{(L+K)!}} P_L^K(\mu') \cos K\phi' \tag{17b}$$

$$\phi_{S,L}^{K}(r,E') \equiv \frac{1}{4\pi}\int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' \psi(r,E',\mu',\phi') \sqrt{\frac{2(L-K)!}{(L+K)!}} P_L^K(\mu') \sin K\phi' \qquad (17c)$$

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. (16) and (17), the odd moments (K+L=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

$$SS = \int_{E'} dE' \sum_{n=1}^{NM} (2L+1)\,\sigma_S^L(r,E' \rightarrow E)\, R_n(\underline{\Omega})\, \tilde{\phi}_n(r,E'), \qquad (18)$$

where NM is the total number of spherical harmonics (and flux moments) required for a given Legendre expansion order, L_o (as shown in Table 12.2), 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 $R_n(\underline{\Omega})$. ISCT is the input variable for L_o, describing the expansion order of the fluxes and the scattering source. The index L in Eq. (18) is the subscript of the Legendre function P_L^K appearing in the appropriate $R_n(\underline{\Omega})$, $0 \le L \le ISCT$. The $R_n(\underline{\Omega})$ are listed in Table 12.3 for typical Legendre expansion orders. For each $R_n(\underline{\Omega})$ in the table, there is a corresponding flux moment defined by Eq. (17a), (17b), or (17c) as appropriate.

Table 12.2 Number of Spherical Harmonics, N, as a Function of Order L_o

L_o	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

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

Table 12.3 Spherical Harmonics, $R_n(\underline{\Omega})$, for Different Geometries

n	Standard Plane and Spherical Geometries P_5[a]	Cylindrical Geometry P_4	Two-Angle Plane P_3
1	$P_o(\mu)$	$P_o(\xi)$	$P_o(\mu)$
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	$P_3(\mu)$	$\frac{\sqrt{3}}{6}P_2^2(\xi)\cos 2\phi$	$P_1^1(\mu)\sin\phi$
5	$P_4(\mu)$	$\frac{\sqrt{6}}{6}P_3^1(\xi)\cos 2\phi$	$P_2(\mu)$
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_1 denotes L-th order Legendre Expansion.

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 12.3. First, the inhomogeneous source moments are defined for a Legendre expansion order IQAN:

$$Q_L(r, E) \equiv \frac{1}{4\pi} \int_{-1}^{1} du \int_{0}^{2\pi} d\phi Q(r, E, \underline{\Omega}) P_L(\mu), \qquad L = 0, ..., IQAN, \tag{19a}$$

$$Q_{C,L}^K(r, E) \equiv \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi Q(r, E, \underline{\Omega}) P_L^K(\mu) \cos K\phi, \text{"'} \tag{19b}$$

$$L = 0, ..., IQAN \qquad K = 1, ..., L ,$$

$$Q_{S,L}^K(r, E) \equiv \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi Q(r, E, \underline{\Omega}) P_L^K(\mu) \sin K\phi \tag{19c}$$

The inhomogeneous source is represented in the general spherical harmonic expansion

$$Q(r, E, \underline{\Omega}) = \sum_{n=1}^{NMQ} (2L+1) R_n(\underline{\Omega}) \tilde{Q}_n(r, E), \tag{20}$$

where NMQ is the total number of spherical harmonics (and source moments) required for a given Legendre expansion order, L_o, as shown in Table 12.2, 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 the $R_n(\underline{\Omega})$. IQAN is determined from the number of spherical harmonics, NMQ, in the input. The index L is the subscript of the Legendre function P_L^K appearing in the appropriate $R_n(\underline{\Omega})$, $0 \le L \le IQAN$. The $R_n(\underline{\Omega})$ are listed in Table 12.3 for typical Legendre expansion orders. For each of these $R_n(\underline{\Omega})$ is a corresponding source moment defined by Eqs. (19a), (19b), or (19c), as appropriate.

Discrete-Ordinates Equations in One Dimension

The number of discrete ordinates or angles in the various one-dimensional symmetries depend upon the symmetry and the S_n order desired. In Table 12.4 is given the number of angles used in each case.

Table 12.4 Number of Quadrature Points, M as a Function of S_n Order, N

N	Standard Plane Geometry	Two-Angle Plane Geometry	Cylindrical Geometry	Spherical Geometry
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

a. Standard Plane Geometry

For standard plane geometry (see Table 12.1 and Figure 12.1) 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 Figure 12.4. Note that the weights, w_m, correspond to $d\mu_m/2$ for this geometry. The angular flux moments, given by Eq. (17a), are approximated by

$$\phi_L(x) \cong \sum_{m=1}^{MM} w_m P_L(\mu_m) \psi_m(x), \tag{21}$$

The discrete-ordinates approximation to the transport equation, i.e., Eq. (3), becomes

$$\mu_m \frac{\partial \psi_m(x)}{\partial x} + \sigma(x) \psi_m(x) = S_m(x) \tag{22}$$

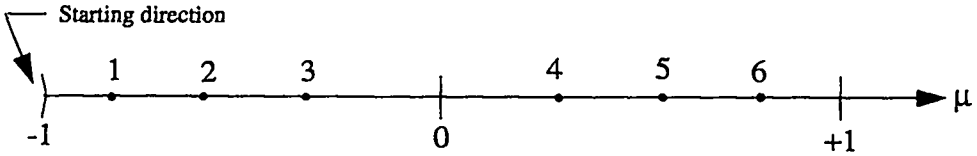


Figure 12.4 Ordering of S_6 directions in plane and spherical geometries

Note: The starting direction only applies to spherical geometry

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 Figure 12.5. The weights, w_m, correspond to $d\Omega_m/4\pi$ for this option. The angular flux moments, given by Eqs. (17a)-(17c), are approximated by

$$\phi_L(x) \equiv \sum_{m=1}^{MM} w_m P_L(\mu_m)\,\psi_m(x), \tag{23a}$$

$$\phi^K_{C,L}(x) \equiv \sum_{m=1}^{MM} w_m \psi_m(x)\, P^K_L(\mu_m)\cos K\phi_m, \tag{23b}$$

$$\phi^K_{S,L}(x) = \sum_{m=1}^{MM} w_m \psi_m(x)\, P^K_L(\mu_m)\sin K\phi_m \tag{23c}$$

The discrete-ordinates approximation to the transport equation is the same as for standard plane geometry, i.e., Eq. (22) on page 12-27.

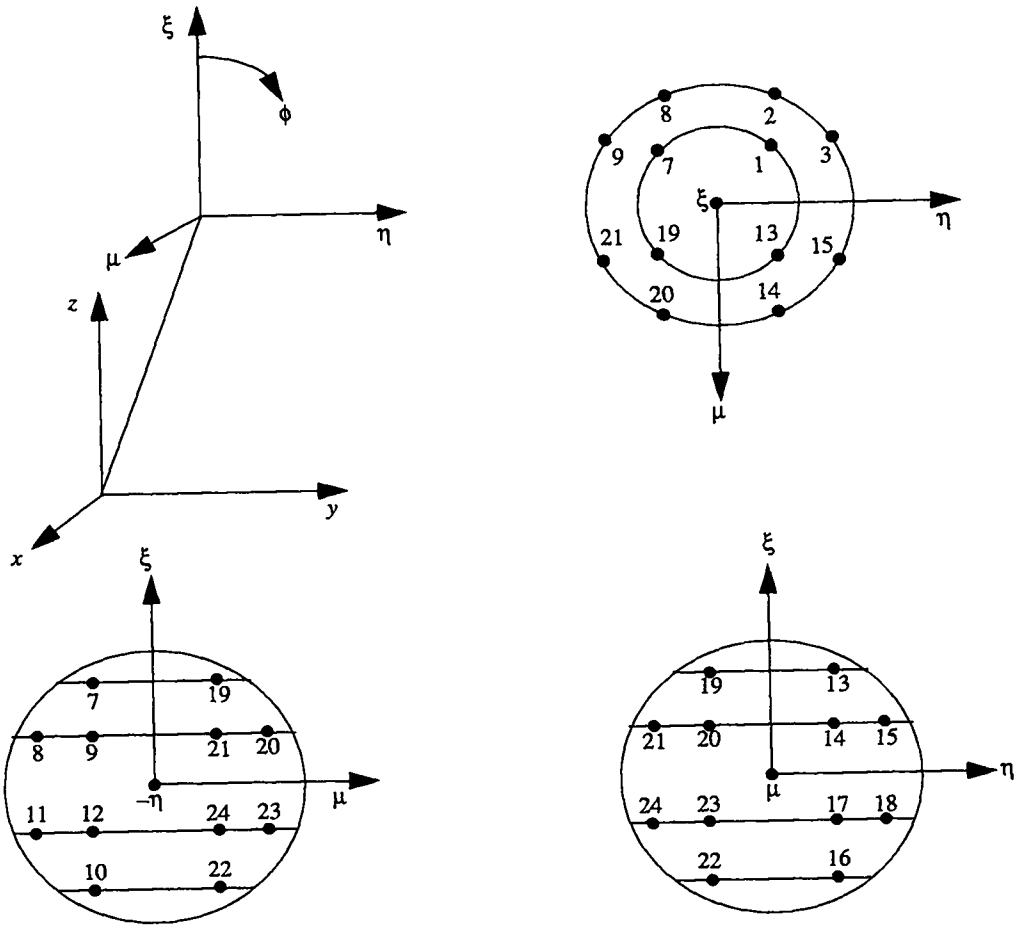


Figure 12.5 Ordering of S_4 directions in two-angle plane geometry*

c. Cylindrical Geometry

For cylindrical geometry (see Table 12.1 and Figure 12.2), the multigroup transport equation, Eq. (3), may be written

$$\mu\frac{\partial(r\psi)}{\partial r} - \frac{\partial(\eta\psi)}{\partial\phi} + r\sigma\psi = rS(r,\underline{\Omega}), \tag{24}$$

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 associ-

*The ordinates in the octant $\mu, \xi < 0, \eta > 0$ are not shown.

ated quadrature weights w_m. The ordering of these quadrature points is illustrated in Figure 12.6 for an S_6 quadrature.

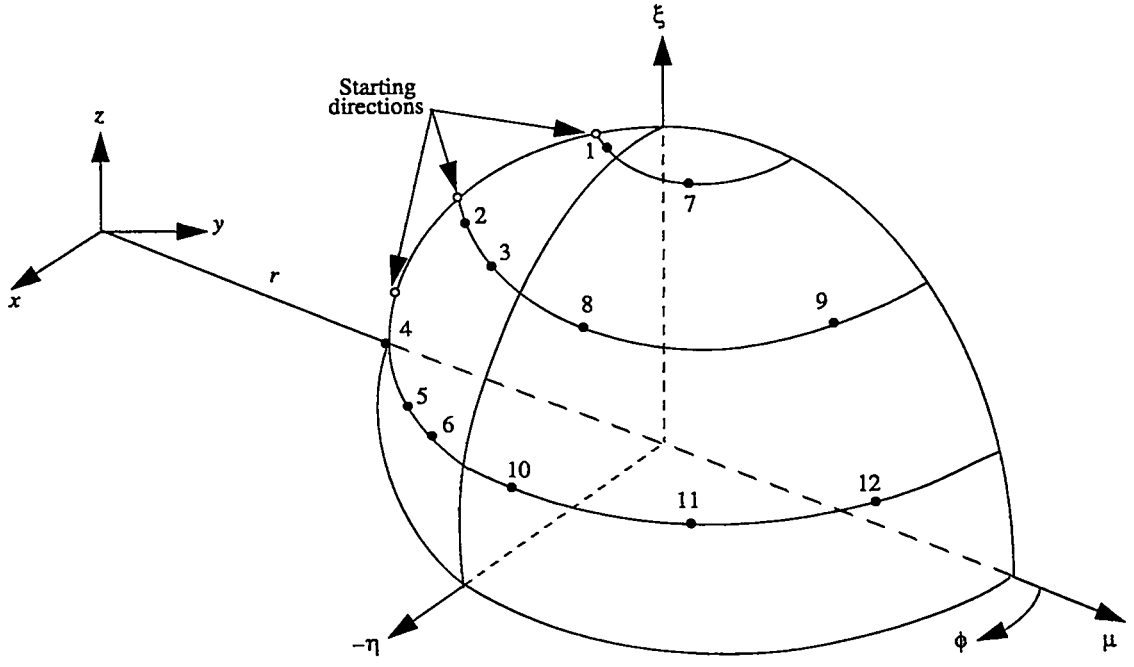


Figure 12.6 Ordering of S_6 directions in cylindrical geometry

As before, $\psi_m(r) \equiv \psi(r, \mu_m \eta_m)$ represents the average angular flux in $d\Omega_m$ about Ω_m and the angular flux moments for direction m are given by Eqs. (23a)-(23b). In addition, it is necessary to define angular-cell-edge fluxes on a given ξ-level as $\psi_{m-1/2}(r)$ and $\psi_{m+1/2}(r)$. The discrete-ordinates approximation to Eq. (24) can then be written:

$$\mu_m \frac{\partial(r\psi_m)}{\partial r} + \left(\frac{\alpha_{m+1/2}}{w_m}\right)\psi_{m+1/2}(r) - \left(\frac{\alpha_{m-1/2}}{w_m}\right)\psi_{m-1/2}(r) + r\sigma\psi_m(r) = rS_m(r) \quad (25)$$

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

$$\alpha_{m+1/2} - \alpha_{m-1/2} = -w_m \mu_m, \tag{26}$$

with the requirement that the first $(\alpha_{1/2})$ and last $(\alpha_{M+1/2})$ coefficients on each ξ-level must vanish. It can be shown that Eq. (25) becomes identical to Eq. (24) in the limit of vanishingly small angular intervals. In the output of ONEDANT pertaining to the angular quadrature, the quantities $\left(\frac{\alpha_{m+1/2}}{w_m}\right)$ and $\left(\frac{\sigma_{m-1/2}}{w_m}\right)$ are printed out under the headings BETA PLUS and BETA MINUS, respectively.

d. Spherical Geometry

From Table 12.1 the multigroup transport equation, Eq. (3), can be written

$$\mu \partial \frac{(r^2\psi)}{\partial r} + r\frac{\partial[(1-\mu^2)\psi]}{\partial \mu} + r^2\sigma\psi = r^2 S(r,\mu), \tag{27}$$

where azimuthal symmetry in ϕ (see Figure 12.3) has been assumed. The angular domain $\mu \in [-1, 1]$ is discretized into MM 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 Figure 12.4. As before, $\psi_m(r) \equiv \psi(r, \mu_m)$ represents the average angular flux in $d\Omega_m (= d\mu_m)$ about $\underline{\Omega}_m$ and the angular flux moments, given by Eq. (17a), are approximated by Eq. (16). In addition, it is necessary to define angular-cell-edge fluxes $\psi_{m-1/2}(r)$ and $\psi_{m+1/2}(r)$. The discrete-ordinates approximation to Eq. (27) is then written as

$$\mu_m \frac{\partial\left(r^2\psi_m\right)}{\partial r} + \left[\left(\frac{\beta_{m+1/2}}{w_m}\right)\psi_{m+1/2}(r) - \left(\frac{\beta_{m-1/2}}{w_m}\right)\psi_{m-1/2}(r)\right] r + r^2\sigma\psi_m(r) = r^2 S_m(r), \tag{28}$$

where the angular coupling coefficients β must satisfy the recursion relation

$$\beta_{m+1/2} - \beta_{m-1/2} = -2w_m\mu_m, \qquad m = 1, ..., MM, \tag{29}$$

with the requirement from particle conservation that the first $(\beta_{1/2})$ and last $(\beta_{MM+1/2})$ coefficients must vanish. It can be shown[1] that Eq. (28) becomes identical to Eq. (27) in the limit of vanishingly small angular intervals. In the output of ONED-

ANT pertaining to the angular quadrature, the quantities $\left(\frac{\beta_{m+1/2}}{2w_m}\right)$ and $\left(\frac{\beta_{m-1/2}}{2w_m}\right)$ are printed out under the headings BETA PLUS and BETA MINUS, respectively.

e. Starting Directions

For the curved geometries discrete-ordinates Eqs. (25) and (28), there are three variables to be determined at each space position, r: the angular-cell-edge fluxes $\psi_{m-1/2}(r)$ and $\psi_{m+1/2}(r)$ and the average angular flux $\psi_m(r)$. The $\psi_{m-1/2}(r)$ flux can be assumed known (except for $\psi_{1/2}(r)$) from the previous angular mesh-cell computation and assuming continuity at the angular mesh-cell boundaries. The standard diamond-difference[2] assumption in angle is made to relate the $\psi_{m+1/2}$ to ψ_m, namely,

$$\psi_m(r) = \frac{1}{2}[\psi_{m-1/2}(r) + \psi_{m+1/2}(r)]. \tag{30}$$

Using Eq. (30) to solve for $\psi_{m+1/2}$ and substituting the resulting expression into Eq. (25) or (28), there remains but one equation for the one unknown $\psi_m(r)$.

The assumption that $\psi_{m-1/2}$ 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 $\psi_{1/2}(r)$. For spherical geometry this starting direction is the straight-inward direction $\mu = -1$ for which the term $\left(1-\mu^2\right)\psi$ in Eq. (27) vanishes. This yields a special form of Eq. (28) which can be solved for $\psi_{1/2}(r)$. For cylindrical geometry, as shown in Figure 12.6, starting directions corresponding to ordinates directed towards the cylindrical axis, $\eta = 0, \varphi = 180°$, are used for each ξ-level to yield special equations for $\psi_{1/2}(r)$ on each ξ-level. At the origin, r=0, we also impose an isotropic condition on the angular flux; i.e., $\left.\frac{\partial\psi}{\partial\mu}\right|_{r=0} = 0$. This implies that $\psi_m(0)$ = constant for all m. The constant is determined by solving the starting direction equation as $\frac{d\psi}{dr} + \sigma\psi = S$ which applies to spherical geometry and to cylindrical geometry on each ξ level.

Discretization of the Spatial Variable

The spatial domain of the problem is ultimately partitioned into IT fine-mesh intervals of width Δx_i, $i = 1, 2, ..., IT$ such that $\Delta x_i \equiv x_{i+1/2} - x_{i-1/2}$. Subscripts with half-

integer values denote interval boundaries, and integer subscripts denote interval average, or midpoint, values. It is assumed that $x_{i+1/2} > x_i > x_{i-1/2}$. 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.

The spatial discretization of the one-dimensional symmetries is begun by integrating the Eqs. (22), (25), or (27) for slabs, cylinders, and spheres respectively over a spatial mesh cell. The resulting equation is called the discretized balance equation and is the one solved by the ONEDANT code. This balance equation for all 3 symmetries is written in the following form:

$$\mu_m \left(A_{i+1/2}\psi_{g,m,i+1/2} - A_{i-1/2}\psi_{g,m,i-1/2}\right) +$$

$$+ \left(A_{i+1/2} - A_{i-1/2}\right) \frac{\left(\alpha_{m+1/2}\psi_{g,m+1/2,i} - \alpha_{m-1/2}\psi_{g,m-1/2,i}\right)}{w_m} +$$

$$+ \sigma_{t,g,i} V_i \psi_{g,m,i} = S_{g,m,i} V_i \qquad (31)$$

$$m = 1,...,\mathrm{MM}; \quad i = 1,...,\mathrm{IT}$$

where,

$\psi_{g,m,i}$ is the spatial cell centered angular flux,

$\psi_{g,m,i+1/2}$ is the cell edge angular flux at the right cell edge,

$A_{i+1/2}$ is the area of the cell at the right edge, and

V_i is the volume of cell i.

Equation (31) represents IT*MM equations for 3*IT*MM+MM+IT unknowns for each group. The boundary conditions give an additional IT+MM equations. To generate the remainder of the equations, we resort to an approximation which is called diamond differencing. In this case, we specify a relationship between the cell centered and cell edge fluxes in the following way:

$$\psi_{g,m,i} = 0.5\left(\psi_{g,m,i+1/2} + \psi_{g,m,i-1/2}\right)$$

$$\psi_{g,m,i} = 0.5\left(\psi_{g,m+1/2,i} + \psi_{g,m-1/2,i}\right) \qquad (32)$$

$$m = 1,...,\mathrm{M}; \quad i = 1,...,\mathrm{IT}$$

These equations give the needed 2*IT*MM relationships needed to solve the discretized transport equation. The solution process starts from a known boundary condition which specifies the edge flux at that boundary and follows the particle flow; Eqs. (32) are used to eliminate the unknown edge flux in terms of the known edge and cell centered fluxes.

This is then substituted into Eq. (31) to derive an equation for the cell centered flux. Eq. (32) is then used to compute the value for the unknown edge flux from this cell centered flux and the known edge flux. However, for a variety of reasons, the most common of which is that the cells are too large in terms of mean-free-paths, the value for the edge flux can extrapolate to a negative value. This is unphysical given that the source is positive, thus, a fixup is employed. This negative flux fixup sets the unknown edge flux to zero when it extrapolates to a negative value. The balance equation, Eq. (31), is then resolved under this condition in order to maintain particle balance. Thus this scheme is known as diamond differencing with set-to-zero fixup and is the ONEDANT basic method for spatial differencing.

TWODANT METHODS

The basic discretization methods used in TWODANT are the same as in ONEDANT with the natural extensions to two dimensions. The geometries treated are planar X-Y and polar R-Θ, and cylindrical R-Z, and these are subsets of the symmetries presented in Figs. 12.1 and 12.2 in the ONEDANT section. In the following paragraphs we present some of the details specific to the TWODANT module and the methods we use in two dimensions.

Some Angular Details in TWODANT

The angular variable is treated using discrete ordinates as in ONEDANT, but the domain is extended to the hemisphere due to the two-dimensional geometries. The explicit form used in the spherical harmonics expansion of the sources is given in Table 12.5.

Table 12.5 Spherical Harmonics in Two Dimensions

n	Two-Dimensional Geometries P_3
1	$P_o(\mu)$
2	$P_1(\mu)$
3	$P_1^1(\mu)\cos\phi$
4	$P_2(\mu)$
5	$\frac{\sqrt{3}}{3}P_2^1(\mu)\cos\phi$
6	$\frac{\sqrt{3}}{6}P_2^2(\mu)\cos 2\phi$
7	$P_3(\mu)$
8	$\frac{\sqrt{6}}{6}P_3^1(\mu)\cos\phi$
9	$\frac{\sqrt{15}}{30}P_3^2(\mu)\cos 2\phi$
10	$\frac{\sqrt{10}}{60}P_3^3(\mu)\cos 3\phi$

This table shows the specific case for P_3 expansion which has 10 moments. In general the number of moments is given by $(L+1)(L+2)/2$ where L is the Legendre expansion order.

In TWODANT we have two possible arrangements of the discrete ordinate points on an octant of the unit sphere. One arrangement is triangular in which the first level gets one point, the second gets two, and so forth. The second is a square arrangement in which all levels get the same number of points equal to the number of levels. A fairly complete discussion of the issues involved, especially the concept of levels is in section IV.A of Ref. 4. In Table 12.6 we show the number of quadrature points as a function of S_n order for the two-dimensional symmetries in the triangular and square arrangements.

Table 12.6 Number of Angles Per Octant in Two Dimensions

N	Triangular Arrangement	Square Arrangement
2	1	1
4	3	4
6	6	9
8	10	16
12	21	36
16	36	64
N	$N(N+2)/8$	$N \cdot N/4$

Transport Operator in Two-Dimensional Symmetries

The transport operator for each of the two-dimensional symmetries that TWODANT treats are listed in the following. They are written in conservative form since this is the form that is used to derive the spatially discretized equations solved in the code.

Planar X-Y Symmetry

$$\underline{\Omega}_m \bullet \nabla\psi_m = \mu_m \frac{\partial \psi_m}{\partial x} + \eta_m \frac{\partial \psi_m}{\partial y} \tag{33}$$

where,

$$\mu_m = \hat{e}_x \cdot \underline{\Omega}_m,$$

$$\eta_m = \hat{e}_y \cdot \underline{\Omega}_m.$$

Planar R-Θ Symmetry

$$\underline{\Omega}_m \bullet \nabla\psi_m = \frac{\mu_m}{r}\frac{\partial}{\partial r}(r\psi_m) + \frac{\eta_m}{r}\frac{\partial\psi_m}{\partial\theta} - \frac{1}{r}\frac{\partial}{\partial\varphi}(\eta\psi_m) \tag{34}$$

where,

$$\mu_m = \hat{e}_r \cdot \underline{\Omega}_m,$$

$$\eta_m = \hat{e}_\theta \cdot \underline{\Omega}_m.$$

Cylindrical R-Z Symmetry

$$\underline{\Omega}_m \bullet \nabla\psi_m = \frac{\mu_m}{r}\frac{\partial}{\partial r}(r\psi_m) - \frac{1}{r}\frac{\partial}{\partial\varphi}(\eta_m\psi_m) + \xi_m\frac{\partial\psi_m}{\partial z} \tag{35}$$

where,

$$\mu_m = \hat{e}_r \cdot \underline{\Omega}_m,$$

$$\xi_m = \hat{e}_z \cdot \underline{\Omega}_m.$$

Spatially Discretized Two-Dimensional Transport Equation

The spatially discretized equations for all the symmetries can be written as a balance equation in a single form. The balance equation is derived by integrating the above equations over a spatial mesh cell.

$$\mu_m (A_{i+1/2,j}\psi_{g,m,i+1/2,j} - A_{i-1/2,j}\psi_{g,m,i-1/2,j}) +$$

$$+ \eta_m B_{i,j} (\psi_{g,m,i,j+1/2} - \psi_{g,m,i,j-1/2}) +$$

$$+ (A_{i+1/2,j} - A_{i-1/2,j}) \frac{(\alpha_{m+1/2}\psi_{g,m+1/2,i,j} - \alpha_{m-1/2}\psi_{g,m-1/2,i,j})}{w_m} +$$

$$+ \sigma_{t,g,i,j} V_{i,j} \psi_{g,m,i,j} = S_{g,m,i,j} V_{i,j} \tag{36}$$

where,

$\psi_{g,m,i+1/2,j}$ is the flux on the right edge of the mesh cell,

$\psi_{g,m,i,j+1/2}$ is the flux on the top edge of the mesh cell,

$\psi_{g,m+1/2,i,j}$ is the angular direction edge flux,

$\psi_{g,m,i,j}$ is the cell center angular flux,

$A_{i+1/2,j}$ is the mesh cell area in the i coordinate direction,

$B_{i,j}$ is the mesh cell area in the j coordinate direction,

$V_{i,j}$ is the mesh cell volume.

We note that in X-Y symmetry, the areas in the *i* direction, $A_{i+1/2,j}$, are all equal to unity, and hence the term with the angular derivative vanishes.

Equation (36) represents IT*JT*MM equations for 4*IT*JT*MM + (IT+JT)*MM + IT*JT unknowns for each group. The boundary conditions give an additional IT*JT + (IT+JT)*MM equations. As implemented in the TWODANT code, we generate the remainder of the equations by one of two approximations: diamond differencing with set-to-zero fixup or adaptive weighted diamond differencing. In the diamond case, we specify a relationship between the cell centered and cell edge fluxes in the following way:

$$\psi_{g,m,i,j} = 0.5\,(\psi_{g,m,i+1/2,j} + \psi_{g,m,i-1/2,j})$$

$$\psi_{g,m,i,j} = 0.5\,(\psi_{g,m,i,j+1/2} + \psi_{g,m,i,j-1/2})$$

$$\psi_{g,m,i,j} = 0.5\,(\psi_{g,m+1/2,i,j} + \psi_{g,m-1/2,i,j}) \qquad (37)$$

$$m = 1,...,\mathrm{MM}; \quad i = 1,...,\mathrm{IT}; \quad \mathrm{j}=1,...,\mathrm{JT}$$

These equations give the needed 3*IT*JT*MM relationships needed to solve the discretized transport equation. The solution process starts from a known boundary condition which specifies the edge flux at that boundary and follows the particle flow; Eqs. (37) are used to eliminate the unknown edge flux in terms of the known edge and cell centered fluxes. This is then substituted into Eq. (36) to derive an equation for the cell centered flux. Eq. (37) is then used to compute the value for the unknown edge flux from the cell centered and the known edge fluxes. However, as in the one dimensional case, the value for the edge flux can extrapolate to a negative value and so a fixup is employed. This negative flux fixup sets the unknown edge flux to zero if it extrapolates negative, and the balance equation, Eq. (36), is resolved under this condition to maintain particle balance.

The second method that we use for the spatial discretization is adaptive weighted diamond differencing (AWDD). This is based upon a weighted diamond approximation which, rather than fixing up, uses a predictor corrector method to determine the weights that will give a positive edge flux in each of the coordinate directions. The weighted diamond equations are written as:

$$(1 + P_{x,g,m,i,j})\,\psi_{g,m,i,j} = \begin{pmatrix} \psi_{g,m,i+1/2,j} + P_{x,g,m,i,j}\psi_{g,m,i-1/2,j} \\ P_{x,g,m,i,j}\psi_{g,m,i+1/2,j} + \psi_{g,m,i-1/2,j} \end{pmatrix} \begin{pmatrix} \mu_m > 0 \\ \mu_m < 0 \end{pmatrix}$$

$$(1 + P_{y,g,m,i,j})\,\psi_{g,m,i,j} = \begin{pmatrix} \psi_{g,m,i,j+1/2} + P_{y,g,m,i,j}\psi_{g,m,i,j-1/2} \\ P_{y,g,m,i,j}\psi_{g,m,i,j+1/2} + \psi_{g,m,i,j-1/2} \end{pmatrix} \begin{pmatrix} \eta_m > 0 \\ \eta_m < 0 \end{pmatrix}$$

$$(1 + P_{a,g,m,i,j})\,\psi_{g,m,i,j} = P_{a,g,m,i,j}\psi_{g,m+1/2,i,j} + \psi_{g,m-1/2,i,j} \qquad (38)$$

$$m = 1,...,MM; \quad i = 1,...,IT; \quad j=1,...,JT$$

$$|P_{x,g,m,i,j}| \le 1, \quad |P_{y,g,m,i,j}| \le 1, \quad |P_{a,g,m,i,j}| \le 1$$

In Eq. (38) the weights P are first specified to be unity which corresponds to diamond difference. The unknowns are then solved for depending upon particle flow and substituted into Eq. (36). A test is done to check positivity and if positivity fails, the P's are recalculated using the just obtained diamond information. We also impose a condition of monotonicity which influences the spatial smoothness of the solution; this is controlled by the input parameter WDAMP in Block-V. This is group dependent and a value of 0 says to use the diamond with set-to-zero fixup described above; a non-zero positive value indicates that the AWDD method is to be used, where the value entered governs the smoothness and accuracy of the solution and is normally between to 1 to 4, 1 being the closest to the diamond solution and 4 being heavily damped toward the step solution. More details on this can be found in Ref. 5.

The diamond difference method gives good results for Keff problems and for a reasonably sized mesh gives good integral values for the system. If the problem is one of deep penetration or shielding or if point-wise values for the fluxes are very important, then we recommend the AWDD method with a damping parameter of from 1 to 2.

Monte Carlo/Discrete Ordinates Hybrid Method

Overview

TWODANT/MC is an extension to the existing S_n code TWODANT. TWODANT/MC solves the neutral particle transport equation by a hybrid method that iteratively couples regions where deterministic (S_n) and stochastic (Monte Carlo) methods are applied.[8] Unlike previous hybrid methods, the Monte Carlo and S_n regions are fully coupled in the sense that no assumption is made about geometrical separation or decoupling. TWODANT/MC provides a new means of solving problems involving both optically thick and optically thin regions that neither Monte Carlo nor S_n is well suited for by themselves. TWODANT/MC is capable of solving forward, inhomogeneous source problems in X-Y and R-Z geometries.[9] This capability includes multigroup problems involving upscatter and fission.[10,11]

The fully coupled Monte Carlo/S_n technique used in TWODANT/MC consists of defining spatial and/or energy regions of a problem in which either a Monte Carlo calculation or an S_n calculation is to be performed. The Monte Carlo region may comprise the entire spatial region (with vacuum or reflective boundary conditions) for selected energy groups, or may consist of a rectangular area that is either completely or partially embedded in an arbitrary S_n region. The Monte Carlo and S_n regions are then connected through the common angular boundary fluxes and scattering/fission sources, which are determined iteratively using an response matrix technique.

The hybrid method has been implemented in TWODANT/MC by adding special-purpose vectorized Monte Carlo subroutines, and linkage subroutines to carry out the interface flux iterations. The common angular boundary fluxes are included in TWODANT as interior boundary sources, leaving the logic for the S_n solution of the transport flux unchanged, while the diffusion synthetic accelerator remains effective in accelerating the S_n calculations. The physical description of the Monte Carlo region is derived from the standard TWODANT input file, reducing the required user input at some expense in generality, since this limits the Monte Carlo region to two-dimensional X-Y and R-Z grids. The S_n cross sections are used to form multigroup cross sections for the Monte Carlo region.

TWODANT/MC Input

The additional input required for TWODANT/MC is all located in Block-V of the standard TWODANT input (see page 3-61 through page 3-63). The input format follows all standard DANTSYS input rules as described in the User's Guide. The keyword MCOPT controls whether a given run will be a standard S_n only run (MCOPT=0) or a hybrid Monte Carlo/S_n run (MCOPT=1). A hybrid run is performed only when MCOPT is present and assigned the value one. Otherwise, the TWODANT/MC keywords are disregarded.

The Monte Carlo region

The keywords MCIBND, MCJBND, MCREG, and MCBLT control the size and location of the Monte Carlo region. MCIBND designates the left and right boundaries of Monte Carlo region, where the numerical values represent an S_n fine mesh cell. Thus, if IT=3, MCIBND=1 3 designates the entire horizontal width of the problem geometry as the Monte Carlo region. An input of zero automatically sets the boundary to the left or right cell, respectively. Similarly, MCJBND designates the bottom and top cells of the Monte Carlo region. Note that only a single Monte Carlo region can be designated.

Usually, the Monte Carlo method will be used in an optically thin region, such as a void, and the designated Monte Carlo boundaries will be those of the optically thin region. However, experience has shown that the hybrid Monte Carlo/S_n hybrid method works best when the MC/S_n interface is located in a region where the angular flux is approximately isotropic. Such is not the case at an interface between optically thick and thin regions. Instead, we recommend that the actual interface be located at least one mean free path into the optically thick region. However, since the mean free path can be strongly group dependent, then the size of the recommended "boundary layer" is also group dependent.

To prevent having to require the user to input (and calculate) group dependent region boundaries, we instead use the keyword MCBLT to indicate the number of mean free paths to be added to the designated Monte Carlo region. The code will then automatically calculate the mean free path for a given energy group and material, and extend the actual Monte Carlo/S_n boundary the appropriate the number of fine mesh cells to provide at least an MCBLT sized interface region. To maintain a rectangularly shaped Monte Carlo region, only a single value for the mean free path is used along each face of the designated Monte Carlo region, even if multiple materials are present along a given face. The calculated group-dependent Monte Carlo boundaries are listed in the Monte Carlo Setup Information in the output. Note that the default value of MCBLT is one.

The Monte Carlo region may consist of any or all energy groups in a problem. It is usually best to use Monte Carlo in only the source and/or higher energy groups, since Monte Carlo can become very expensive in lower energy groups where scattering usually predominates. The Monte Carlo energy regions are designated by the keyword MCREG. Note that by assigning all energy and spatial regions to the Monte Carlo region, it is possible to run solely a Monte Carlo calculation with TWODANT/MC.

Source Definition

TWODANT/MC is capable of treating any input fixed source that standard TWODANT is, including the first-collision source. The source(s) may be arbitrarily split between the Monte Carlo and S_n regions. The calculated fraction of the fixed source located in the Monte Carlo region (including boundary layers) is printed out in the Monte Carlo Setup Information section of the output.

In addition, TWODANT/MC is also capable of treating many singular sources that are difficult to use with standard S_n. These source types include point sources, beam sources, and point beam sources. For a complete description, see page 3-63.

Number of Histories and Code Flow

If there is some fixed source located inside the Monte Carlo region, the code will sample it before attempting to do any S_n calculations. The number of histories used to sample the source inside the Monte Carlo region will be MCNHIS*MCNTR (the number of histories/trial times the number of trials). TWODANT/MC runs histories in a fixed number of histories/batch in order to obtain variance estimates for several quantities. While running the batches (trials), the code will print a line on the screen upon the completion of each batch.

In sampling the fixed source, the Monte Carlo will track particles from the fixed source inside the Monte Carlo region through the Monte Carlo region until a particle either crosses into the S_n region, at which time a Monte Carlo to S_n boundary flux will be scored, or until it exits the problem geometry.

Once the fixed source calculation has been completed, the code will proceed on to do a complete S_n calculation for the entire problem geometry as normal (i.e., no Monte Carlo option), except that the Monte Carlo calculated boundary flux at the S_N/Monte Carlo interface will be used as an internal boundary condition. For fixed source calculations with no upscatter or fission, this will consist of one outer iteration. For problems with upscatter and/or fission, outer iterations will be performed until the appropriate S_n convergence criteria have been met. The S_n will print monitor lines to the screen as per normal S_n calculations.

After the S_n calculation has completed, the code will use Monte Carlo to sample the incoming angular boundary fluxes into the Monte Carlo region from the S_n region, tracking particles through the Monte Carlo region as described above. The number of histories used for this calculation is explained below.

Let the magnitude of the incoming current (i.e., leakage) into the Monte Carlo region from the S_n equal TAPRTCLS, and let the magnitude of the fixed source inside the Monte Carlo region be ANORMMC. Then, the total number of histories the code sets out to run in sampling the incoming boundary fluxes is MCNHIS*MCNTR*TAPRTCLS/ANORMMC, divided into MCNTR batches for error estimates. However, the code always checks to see that this number is no greater than MCNHIS*MCNTR. If so, it resets it to MCNHIS*MCNTR to ensure that it doesn't inadvertently run a excessively large number of histories. Again, TWODANT/MC will print a line to the screen after running each batch of histories.

However, TWODANT/MC will always run at least one particle per phase space cell per batch when sampling the boundary fluxes and scattering/fission sources. The number of phase space cells along the S_n/Monte Carlo interface is: the number of spatial (fine mesh) cells along the interface times the number of Monte Carlo energy groups times the number of quadrature directions/quadrant times two plus the number of spatial cells in the Monte Carlo region with scattering/fission sources from the S_N region times the number of Monte Carlo energy groups times the number of quadrature directions/quadrant times four. Thus, regardless of the value calculated above, if P is the number of phase space cells, the code will run at least P*MCNTR number of histories in sampling the boundary fluxes and scattering/fission sources. If P is large, or MCNTR is large, the

code will always run a large number of histories, regardless of the value of MCNHIS, or the size of TAPRTCLS. A quick way to tell how TWODANT/MC is picking the number of histories to run is to look at the average source particle weight in the Link Monitor Table. If the average weight is much less than one, than the number of histories used is being driven by the size of P, not MCNHIS.

The combination of a complete S_n calculation with Monte Carlo sampling of interface boundary fluxes is referred to as a "super-outer" iteration. The convergence criteria for super-outers will be discussed later. On succeeding super-outers, however, the number of histories is determined as described above. Ideally, as the S_n/Monte Carlo boundary fluxes converge, TAPRTCLS decreases, and less histories will be used.

If there is no fixed source inside the Monte Carlo region, the code proceeds directly to a S_n calculation after processing the input deck as usual. Once the S_n calculation is completed, the boundary fluxes are sampled as above, except that since ANORMMC is zero, we determine the number of histories somewhat differently. If ALPHA(1) is the "fit coefficient" of the incoming leakage for the first super-outer, the number of histories used in the (n)th super-outer is MCNHIS*MCNTR*ALPHA(n)/ALPHA(1). (See below for a description of fit coefficients.) Again, however, the same caveats about the size of P and MCNTR apply here.

Convergence Criteria

For a given S_n calculation in a super-outer, standard S_n convergence criteria are used. The convergence criteria for the S_n/Monte Carlo iterations is based upon the errors in the group-dependent scalar boundary fluxes at the S_n/Monte Carlo spatial interface and the group-dependent scalar scattering/fission sources. Presently, the same error criteria value (i.e., EPSI) is used for both the S_n and S_n/Monte Carlo iterations. The desired goal in TWODANT/MC to achieve is a relative error in the Monte Carlo to S_n energy-dependent scalar boundary flux and the Monte Carlo to S_n scalar scattering/fission sources of less than EPSI, when comparing from one super-outer to the next. The error in the outgoing boundary flux and scattering/fission source, where the error is the relative error between the current and previous iteration's fluxes/sources, is determined after the Monte Carlo calculation in a super-outer, and is printed out in the link monitor as the "iteration error." The group location of the maximum errors as well as a cell index are also printed out.

Unfortunately, due to random fluctuations when performing Monte Carlo calculations, it would be almost impossible to meet typical S_n convergence criteria of 10^{-3} or 10^{-4} in the super-outers. Instead, we attempt to achieve "shape convergence" in the incoming boundary fluxes (and sources) from the S_n region to the Monte Carlo region. This is described in detail in Ref. 12. Essentially, the incoming boundary fluxes are decomposed into orthogonal "basis vectors" (shapes), with one new shape per super-outer. The latest basis vector is then sampled via Monte Carlo to obtain an associated "response vector." The response vectors are then used to form an outgoing boundary flux from the Monte Carlo region to the S_n, which is used by the S_n in the next super-outer as an internal boundary condition. (This also saves significantly on storage, since we do not have to calculate an explicit response matrix, which would be approximately of size P^2.)

The idea is that the S_n acts as a "smoothing operator," reducing the random fluctuations from the Monte Carlo operator, so that the incoming boundary fluxes and sources will converge faster. This convergence is seen in the Link Monitor table, which prints outs the "fit coefficients," or the expansion coefficients of the incoming boundary fluxes and sources in terms of the existing basis vectors. The error printouts labeled "fit error" immediately above the fit coefficients are not the error in the incoming boundary fluxes/ sources from one super-outer to the next, but rather the error in the fit of the basis vectors to the actual incoming boundary flux.

Ideally, as we progress in super-outers, the magnitude of a given shape (or basis vector) goes to a constant, while the relative magnitude of subsequent basis vectors in comparison to earlier ones is monotonically decreasing. Eventually, we have enough shapes to describe the incoming boundary flux to within EPSI, and have thus achieved shape convergence, even though the relative error in the outgoing boundary fluxes may still be much greater than EPSI. At this point, TWODANT/MC uses some simple linear algebra to project what the outgoing boundary flux should be, and runs at least more super-outer (S_n calculation only) to check the projected solution by attempting to converge the outgoing interface fluxes and sources. If it really did have enough shapes to describe the boundary fluxes, the errors in both the incoming and outgoing fluxes (and sources) after projection should meet the convergence criteria, or be very close to it. (Note that since TWODANT/MC does not form any new shapes on the super-outer it projects or the ones to check the projection, it does not require any further Monte Carlo calculations.) If convergence of the outgoing boundary fluxes is not achieved within MCITS iterations, a caution message is printed to the screen and output file.

For problems with large Monte Carlo/S_n interfaces, it usually very difficult to describe all cells along the space/energy boundary to the desired error criteria. Thus, we also use a "global convergence" criteria to determine when shape convergence has been reached. If the relative magnitudes of the smallest two calculated fit coefficients, after division by the magnitude of the largest fit coefficient, are both less than EPSI, than we consider the MC/S_n interface fluxes and sources to be globally converged, and project the existing based on the existing shapes.

Occasionally, before shape convergence is reached, a point will be reached in the iteration process where the S_n is no longer able to sufficiently smooth out the fluctuations in the Monte Carlo operator, resulting in essentially random shapes. TWODANT/MC has two checks for this: a check to ensure that the magnitude of the newest shape (TAPRTCLS) is decreasing from one super-outer to the next, and a check to ensure that the magnitudes of the individual fit coefficients for a given super-outer's incoming boundary flux are monotonically decreasing. If not, it throws away the last basis vector, and projects on the remaining ones. That is, if TWODANT/MC detected a problem in super-outer four, it would throw away the third basis vector calculated in super-outer three, and project on the remaining two.

Since the first super-outer's shape consists of just the incoming boundary flux and sources to the Monte Carlo region, it is entirely positive. However, because succeeding shapes are constructed through a Gram-Schmidt orthogonalization process, they can be composed of negative elements as well as positive ones. This is reflected in the two quantities TAPRTCLS and APRTCLS, which are essentially the magnitudes of the absolute value of the boundary flux/source and the actual boundary flux/source, respectively.

To sample negative elements of the boundary flux and source, TWODANT/MC allows Monte Carlo histories with negative weights. Ideally, the magnitude of any negative elements is small enough so as not to cause any significant problems. Sometimes, however, the tracking of negative particles will result in a net negative angular flux (or source) for an outgoing boundary flux spatial cell. If so, we set the angular flux to zero and adjust the remaining angular fluxes so as to conserve particles. If the total scalar flux for that cell is negative, TWODANT/MC sets all angular fluxes to zero and prints out warning messages to the screen and output file.

Finally, if MCITS super-outers have been performed without achieving either shape convergence or convergence in the outgoing boundary fluxes, TWODANT/MC ceases iterating, uses the existing shapes for edit calculations, and prints out warning messages to the screen and output file.

Memory Requirements

To use the S_n/Monte Carlo option, additional temporary LCM storage is needed to read in the cross sections for the Monte Carlo, which is done in subroutine MCXSPT, and approximately 2*P*MCITS permanent storage is needed for the basis and response vectors, which is done in subroutine MCXS. Other additional storage is needed for various arrays stored in SCM and LCM, but the shape vectors are generally the predominant ones. TWODANT/MC prints out storage information in the Monte Carlo Setup Information table, and writes the shape vectors to disk if MAXLCM is exceeded.

Treatment of Angular Scattering

The Monte Carlo calculations are all performed using a multigroup treatment, so that no additional cross-section input is required. The treatment of angular scattering is somewhat complicated, however, because the direct sampling of the group-to-group transfer moments is usually not possible since they may be negative. Instead, TWODANT/MC attempts to generate 32 equiprobable bins such that the L Legendre moments of the angular scattering cross sections are conserved. This approach (the maximum entropy method) is further described in Ref. 13. If TWODANT/MC encounters any problems in generating the bins from the cross-section moments, it prints out various error messages which will appear in the Monte Carlo Setup Information Table. Generally, it is best to preserve as many moments as possible when forming the equiprobable bins, so the use of transport corrected cross sections in TWODANT/MC should be avoided.

Variance Reduction

In the hybrid Monte Carlo/S_n method, each technique is used where it is most advantageous. Therefore, extensive variance reduction techniques for the Monte Carlo should not be necessary. However, TWODANT/MC does include the capability for several simple variance reduction and biasing techniques, such as weight cutoffs, splitting, and Russian Roulette (both in space and energy).

To use weight cutoffs, assign two values (w_1 and w_2) to the keyword MCWC (e.g., MCWC= .10 .05). When the weight of a particle in proportion to its source weight (*w/*

w_s) becomes smaller than w_2, Russian roulette is played, with a probability of survival w/w_s times $1/w_1$. If the particle survives, it is assigned weight $w_1 w_s$. Weight cutoffs are performed after a collision event.

For geometry splitting/Russian roulette, the keyword MCCMIMP is used to assign a cell importance r to each coarse mesh, including cells entirely in the S_n region. The importances are assigned via an (IM,JM) sized array. When crossing into cell j from cell i, a particle is assigned the relative importance $R_{ij} = r_j / r_i$. Then, with probability R_{ij} - Int R_{ij}, Int R_{ij} +1 particles are created; otherwise, Int R_{ij} are created. The weight of all split particles is w/R_{ij} . Note that a cell importance of zero eliminates all particles originating in or entering that coarse mesh.

Similarly, the keyword MCEGIMP is used to assign (NGROUP) energy group importances. Splitting/Russian roulette works as described above. Note that energy group importances of zero can be used to, in effect, set an energy cutoff.

The keyword MCLVIMP is used to set level biasing. When sampling interface fluxes and sources in the super-outer iterations, level biasing allows one to turn on or off certain quadrature levels (not directions). In its current form, level biasing allows one to only turn on or off the sampling of particles according to the quadrature level their starting direction falls within. An entry of one samples particles starting on that level normally, while a zero kills them immediately. Thus, although the same number of histories will be run with or without level biasing, the run time will be reduced with level biasing in approximate proportion to the number of levels set to zero, since histories starting within those levels will not be tracked.

The quadrature levels may be identified from the Key Start S_n Constants in the input printout. A given level corresponds to a fixed value of η. For S_{12}, for example, there are six levels, regardless of whether one is using a triangular or product quadrature set. Level one corresponds to quadrature directions with the largest η value, level two to the next largest, etc. To use level biasing, ISN/2 entries of either one or zero must be made in order to bias the η levels (ISN is the S_n order). For example, in an S_6 calculation, if one wishes to start particles from the surface source in bins corresponding to quadrature directions with the largest η only, one would enter "MCLVIMP= 1 0 0."

To determine the outcome of scattering events, TWODANT/MC uses implicit capture and fission. In implicit capture, a particle always survives a scattering event with a weight multiplier of σ_s/σ_t, as opposed to analog capture, where a particle either survives with its weight unchanged, or is absorbed. Implicit fission is just an extension of implicit scattering to fissionable materials, where the weight multiplier is now $(\sigma_s + \nu\sigma_f)/\sigma_t$. To prevent a series of weight multiplications greater than one, TWODANT/MC uses weight splitting. That is, when the weight multiplier w_m is greater than one, then, with probability w_m- Int w_m, it creates Int w_m+1 particles; otherwise, Int w_m are created. The weight of all split particles is w, where w is the pre-collision weight.

The above variance reduction techniques can be used separately, or in any combination together. Note that when weight cutoffs are used in conjunction with other techniques, the source weight is appropriately adjusted when a particle undergoes splitting/Russian roulette. Input cell, energy group, and level importances are printed out in tables in the

Monte Carlo Setup Information section of the output when used. The amount of weight (per source particles) created/destroyed by splitting/Russian roulette, as well as the number of particle tracks, is printed out in the informational table after each Monte Carlo calculation.

Particles created by splitting are placed into a particle bank for subsequent tracking. The number of particles the bank can hold is initially sized at 640. If the bank fills up, more space is allocated to it, within the limits of MAXLCM. The maximum size the bank actually reaches during a Monte Carlo calculation is printed out in the informational table.

Statistics

TWODANT/MC estimates variances by running MCNTR batches of (approximately) MCNHIS histories/batch. Thus, for each super-outer, the code obtains an estimate of the variance in the calculated response for that super-outer's shape the Monte Carlo is currently sampling. Given the fit coefficients for each particular shape, it can then estimate the variance in various quantities. To do this, TWODANT/MC assumes that the sampling of each super-outer's basis vector is independent, i.e., that the shape of a given basis vector is not dependent on those that preceded it. Technically, this is not true, and a covariance term should be included, which would increase the estimated relative errors. However, given sufficient smoothing by the S_n operator, the covariance term is generally small enough that it can be neglected. The accuracy of the variance estimates has been examined by running a problem several times with different random seeds and it has been determined that, given MCNHIS large enough to adequately sample the problem, and MCNTR large enough to estimate the variance, then the variance estimates are indeed reasonable.

Edits

In TWODANT/MC, the Monte Carlo method tracks particles on the coarse mesh geometry grid, not the fine mesh, although particles crossing the S_N/Monte Carlo interface are scored per fine mesh cell. Flux estimates for the Monte Carlo region are provided through a tracklength estimator, and these fluxes are used to replace the S_n calculated fluxes in the Monte Carlo region prior to any edits. However, since flux estimates are only available on the coarse mesh grid, average values are used for all fine mesh cells in a given coarse mesh. If one wishes finer detail in the Monte Carlo region, therefore, one must refine the coarse mesh structure. Since the Monte Carlo tracks on coarse meshes, this will increase the Monte Carlo run time. Also, as the coarse meshes get smaller, more histories will be required in order to get adequate statistics for the flux estimates. Note that where a coarse mesh cell is split between the Monte Carlo and S_n regions, Monte Carlo-calculated fluxes are used only in the Monte Carlo portion of the cell.

TWODANT/GQ METHODS

The TWODANT/GQ methods follow those of TWODANT with some restrictions. The angular quadrature is exactly the same and we use the same spherical expansions for the scattering and volumetric sources. The symmetries treated, however, are restricted to X-Y and R-Z; the R-Θ option has not been implemented. The greatest difference is in the spatial differencing which is a modified diamond differencing with set-to-zero fixup method. Because the spatial mesh is general quadrilaterals, including triangles, the method has had to be generalized to account for the cell edge orientations with respect to the discrete directions of particle travel. Because the method is based upon diamond differencing, it is generally second order in the integral quantities as long as the mesh is not too severely distorted. What is too severe has to be determined experimentally by the user. To accelerate the source iterations, we use a DSA accelerator which has been generalized to treat arbitrary, logically rectangular quadrilaterals. The details of the spatial discretization and the iteration acceleration impact is presented in Ref. 6.

TWOHEX METHODS

The TWOHEX methods follow those of TWODANT with some basic differences. The multigroup approximation shown on page 12-11 applies as do the basics of the discrete ordinates approximation shown on page 12-12.

The iteration procedure is similar in that there are inner and outer iterations, but in TWOHEX, there is no acceleration of the inners and the outers are accelerated with a Chebychev acceleration of the source. There is no diffusion synthetic acceleration in TWOHEX as opposed to the other modules in this code package.

We use the two dimensional spherical harmonics expansions for the scattering and volumetric sources, so the discussion under "Some Angular Details in TWODANT" on page 12-35 applies. The transport operator shown under "Planar X-Y Symmetry" on page 12-36 is valid for TWOHEX as well.

The angular quadrature is basically the same as the triangular and rectangular arrangements of TWODANT, but is based on the symmetries of a sextant rather than a quadrant.

The greatest difference is in the spatial differencing which is based on alternating inverted triangles and is described in Ref. 7.

THREEDANT METHODS

The methods that we use in THREEDANT are the same as in TWODANT except with the natural extensions to three dimensions. The S_n quadrature is the same except that eight octants are used as opposed to the four quadrants in 2 dimensions. The spherical harmonics used are the same as the two-angle slab entries from ONEDANT. The spatial discretization has the same two options of diamond with set-to-zero fixup and the AWDD methods described above in the TWODANT methods section.

REFERENCES

1. 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.
2. 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, p. 211.
3. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," *Nucl. Sci. Eng.* **64**, 344 (1977).
4. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).
5. R. E. Alcouffe, "An Adaptive Weighted Diamond Differencing Method for Three-Dimensional XYZ Geometry," *Trans. Am. Nuc. Soc.* **68**, Part A, 206 (1993).
6. R. E. Alcouffe, "A Diffusion Accelerated S_n Transport Method for Radiation Transport on a General Quadrilateral Mesh," *Nucl. Sci Eng.* **105**, 191-197, 1990.
7. W. F. Walters, "The TLC Scheme for Numerical Solution of the Transport Equation on Equilateral Triangular Meshes," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 151-165.
8. W. F. Filippone, and R. E. Alcouffe, "The S_N/Monte Carlo Response Matrix Hybrid Method," *Nucl. Sci. Eng.* **100**, p. 209 (1988).
9. R. S. Baker, W. F. Filippone, and R. E. Alcouffe, "The Multigroup and Radial Geometry Formulation of the Monte Carlo/S_N Response Matrix Method," *Nucl. Sci. Eng.* **105**, p. 184 (1990).
10. R. S. Baker, "A Fully Coupled Monte Carlo/Discrete Ordinates Solution to the Neutron Transport Equation," Ph.D. Dissertation, University of Arizona, Tucson, AZ (1990).
11. R. S. Baker, W. F. Filippone, and R. E. Alcouffe, "Extension of the Fully Coupled Monte Carlo/S_N Response Matrix Method to Problems Including Upscatter and Fission," Int. Topical Meeting on Advances in Math., Comp., and Reactor Phy., 5, p. 21.2 3-1, Pittsburgh, PA (1991).
12. W. F. Filippone, R. S. Baker, and R. E. Alcouffe, "The Monte Carlo/S_N Hybrid Method With Orthogonal Interface Basis Vectors," Joint Intl. Conf. on Math. Methods and Supercomputing in Nuclear Applications, Karlsruhe, **2**, p. 714 Germany (1993).
13. R. S. Baker, "XREP Module," Los Alamos National Laboratory internal memorandum X-6:RSB-93-208 (1993).

ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT – Code Structure

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

13

XTM — Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

LIST OF TABLES

OPERATION OF THE CODE SYSTEM

The DANTSYS code package is a modular computer program package designed to solve the time-independent, multigroup discrete ordinates form of the Boltzmann transport equation in several different geometries. It was developed as a modular code package consisting of three types of modules: an Input Module, several Solver Modules, and an 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.

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 the DANTSYS package. 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 DANTSYS.

Language

The programming language is standard FORTRAN 77 as defined by the ANSI standard X3.9-1978.[3] Known exceptions are the occasional use of variable names longer than six characters and the use of in-line comments.

Structure

The code is structured in a form that separates the input and the output (or edit) functions from the main calculational (or solver) sections of the code. A more complete description of the code structure is provided in "CODE PACKAGE STRUCTURE" starting on page 13-9.

Standard Interface Files

DANTSYS 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 DANTSYS: ISOTXS, GRUPXS, GEODST, NDXSRF, ZNATDN, SNCONS, FIXSRC,

RTFLUX, ATFLUX, RZFLUX, RAFLUX, and AAFLUX. File descriptions for these files are provided in Ref. 1.

The following code-dependent, binary interface files are used in DANTSYS: MACRXS, BXSLIB, FISSRC, RMFLUX, AMFLUX, AZFLUX, RZMFLX, AZMFLX, RAFLXM, AAFLXM, UCFLUX, LNK3DNT, SNXEDT, ADJMAC, SOLINP, EDITIT, ASGMAT, and an extended GEODST and the GEOSING file both used by TWODANT/GQ. ASCII code-dependent files produced by or usable by the code package are MACBCD, XSLIBB, XSLIBF, XSLIBE, ARBFLUX, EDTOUT, and EDTOGX. File descriptions for these code-dependent files are provided in the chapter "FILE DESCRIPTIONS" starting on page 15-1.

The use of the above interface files is described in "CODE PACKAGE STRUCTURE" starting on page 13-9 of this chapter.

Data Management and Transfers

DANTSYS is designed with data-management techniques 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 CCCC standardized 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 transfer using random (direct)-access files, the CCCC procedures have been implemented in the DANTSYS package. 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.

Central Memory Restrictions

DANTSYS was originally designed to be operable within a 50,000-word central memory. But now memory is obtained from the heap, and thus is limited only by the size of the heap.

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.)

CODE PACKAGE STRUCTURE

Each code in the DANTSYS 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. There are several Solver Modules, one for each code in the package, but the Input Module and Edit Modules service them all.

The Solver Modules also generate 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.

A complete list of the interface files accepted, used, and generated by the modules is shown in Table 13.1 The table indicates which modules read or write a particular file. In the table, the notation A means always, O means optionally. Thus for the Edit Module, we see an A for the GEODST file indicating that if the Edit Module is invoked, it will always need a GEODST file to read. An entry of A for output is subject to the provision that the information to be put on the file is available. If there is geometry input, then the Input Module will always write a GEODST file. But, if there was no geometry input, the Input Module would not write the GEODST file.

Table 13.1 Files Read and Written

Information Type	File	Input Module		Solver Module		Edit Module	
		Read	Write	Read	Write	Read	Write
Geometry Information	GEODST	O	A	A	O	A	
	Card Images	O					
	GEOSING[a]				A	O	

Table 13.1 Files Read and Written (Cont.)

Information Type	File	Input Module		Solver Module		Edit Module	
		Read	Write	Read	Write	Read	Write
Cross Sections	ISOTXS	O					
	GRUPXS	O					
	MENDF	O					
	MACRXS / ADJMAC	O	A	A			
	MACBCD	O	O				
	XSLIB	O					
	XSLIBE		O				
	XSLIBF		O				
	BXSLIB	O	O				
	XSLIBB	O	O				
	SNXEDT		A			O	
	Card Image	O					
Material Mixing	NDXSRF/ ZNATDN	O	A	A		O	
	Card Images	O					
	LNK3DNT[b]			O		O	
Assignment of Materials to Zones	ASGMAT		A	A	O	O	
	Card Images	O					
Solver Module Input	SOLINP	O	A	A			
	Card Images	O					
Quadrature	SNCONS			O	A		
Inhomoge-neous Sources	FIXSRC			O	O		
	Card Images	O					
Edit Module Input	EDITIT		A			A	
	Card Images	O					

Table 13.1 Files Read and Written (Cont.)

Information Type	File	Input Module		Solver Module		Edit Module	
		Read	Write	Read	Write	Read	Write
Other Output Files	RTFLUX/ ATFLUX			O[c]	A	A	O
	RAFLUX/ AAFLUX				O		
	RZFLUX/ AZFLUX						O
	RMFLUX/ AMFLUX			O[d]	O	O	
	RAFLXM/ AAFLXM				O		
	RZMFLX/ AZMFLX						O[e]
	ARBFLUX				O[f]		
	FISSRC				O		
	EDTOUT						O
	EDTOGX						O
	UCFLUX			O[f]	O[f]		

a. This file is used only in TWODANT/GQ, it is not produced by any other solver.

b. TWODANT and THREEDANT only.

c. Some Solver Modules accept a flux guess from the RTFLUX/ATFLUX file; some do not.

d. Some Solver Modules accept a flux guess from the RMFLUX/AMFLUX file; some do not.

e. Requires an RMFLUX or AMFLUX file from the Solver.

f. From the TWODANT module only.

A segmented structure is used in DANTSYS for implementing the modules. Such a structure involves the use of a main driver together with input, solver, and edit segments.

The main program, DRIVER, controls the calling of the primary segments, together with those service subroutines used by more than one segment.

The first segment constitutes the Input Module. It is structured into a driver routine, INPT10, plus twelve secondary sections as shown in Table 13.2. Each of the secondary sections performs a unique function so that the Input Module itself is constructed in a modular form.

Table 13.2 Structure of the Input Module

Routine	Function
INPT10	Input Module Driver: controls the flow of the code by calling one or more of the secondary somebodies below.
INPT11	Controls code setup and storage allocation.
INPT12	Controls geometry data processing
INPT13	Controls cross-section library processing for XSLIB, MENDF, XSLIBB, and MACBCD library forms.
INPT14	Controls mixing specification processing.
INPT15	Controls GRUPXS cross-section library processing.
INPT16	Controls ISOTXS cross-section library processing.
INPT17	Controls BXSLIB cross-section library processing.
INPT18	Controls Solver Module input data processing.
INPT19	Controls Edit Module input data processing.
INP110	Controls cross-section balancing operation.
INP111	Controls adjoint reversals.
INP112	Controls GEODST file post processing.

The second segment constitutes the Solver, or calculational, module. It consists of a driver routine plus additional secondary sections. The structure of the ONEDANT Solver Module is depicted in Table 13.3, for TWODANT in Table 13.4, and for THREEDANT in Table 13.5.

Table 13.3 Structure of the Onedant Solver Module

Routine	Function
GRND20	Solver Module Driver; controls the flow of the code by calling one or more of the secondary submodules below.
INPT21	Controls module initializations.
INPT22	Controls quadrature selection.
INPT23	Controls flux guess and inhomogeous source processing.
GRND24	Controls calculational data preparation
GRND25	Controls the outer iterations.
OUTT26	Controls final Solver Module printing.
OUT27	Controls binary file preparation.

Table 13.4 Structure of the Twodant Solver Module

Routine	Function
TIGF20	Solver Module Driver; controls the flow of the code by calling one or more of the secondary submodules below.
TINP21	Controls module initializations.
TINP22	Controls flux guess and inhomogeous source processing.
TINP23	Controls quadrature selection.
TINP24	Checks spatial mesh input for consistency.
TGND25	Calculates initially required functions and grid structures; calculates the first collision source and Monte Carlo Source if required, initializes fission source.
TRANSO	Controls the inner iterations.

Table 13.4 Structure of the Twodant Solver Module (Cont.)

Routine	Function
DIFFO	Controls the outer iterations.
LINKO	Controls the S_n/MC iterations.
TOT28	Controls final Solver Module printing.
TOT29	Controls binary file preparation.

Table 13.5 Structure of the Threedant Solver Module

Routine	Function
TIGFA03D	Solver Module Driver; controls the flow of the code by calling one or more of the secondary submodules below.
TINP213D	Controls module initializations.
TINP223D	Controls flux guess and inhomogeous source processing.
TINP233D	Controls quadrature selection.
TINP243D	Checks spatial mesh input for consistency.
TINP253D	Calculates initially required functions and grid structures; initializes the fission source.
TRANSO3D	Controls the inner iteration.
DIFFO3D	Controls the outer iteration.
TOT283D	Controls final Solver Module printing.
TOT293D	Controls binary file preparation.

The third segment is the Edit Module. It currently consists of a driver routine, OUTT30, plus four secondary sections as shown in Table 13.6.

Table 13.6 Structure of the Edit Module

Routine	Function
OUTT30	Edit module driver; controls the flow of the code by calling one or more of the secondary submodules below.
OUTT31	Controls reaction rate calculations.
OUTT32	Controls power normalization, edit zone averaging, and output file preparation.
OUTT33	Controls a spatial mesh collapse determination.
OUTT34	Controls the mass edit request on the coarse mesh or edit zones.

A fourth segment is used in DANTSYS. This fourth segment 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 segment is not considered to be a module in the sense of the first three segments.

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 listed in Table 13.1. As is indicated in the table, 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. These input data are described in the appropriate User's Guide for the code of interest.

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 given in Table 13.1 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.

A second, more complex, example of the function of the Input Module 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 Table 13.1. 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.

Solver Modules

The Solver Modules of DANTSYS have the function of effecting numerical solutions of the multigroup form of the neutral-particle steady-state Boltzmann transport equation. Separate Solver Modules are used for different basic geometries. For instance, there are different solvers for 1-d, 2-d, and 3-d. The discrete-ordinates approximation is used for treating the angular variation of the particle distribution and the diamond-difference scheme[4] or the adaptive weighted diamond difference scheme in 2 and 3-d,[5] is used for phase space discretization.

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, TWODANT, TWODANT/GQ, and THREEDANT Solver Modules employ the diffusion synthetic acceleration method developed by Alcouffe,[6] an extremely effective method for accelerating the convergence of the iterations. The TWOHEX code uses a higher order scheme with Chebyshev acceleration of the outer iterations. A relatively detailed development of the solution methods used in the Solver Modules is provided in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1.

The Solver 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 listed in Table 13.1. 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, RMFLUX or AMFLUX for TWODANT or THREEDANT, 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. If desired by the user, the angular flux moments code-dependent interface file RMFLUX (or AMFLUX, if an adjoint problem were run) would be produced.

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 and multiplied, and the product printed.

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 Table 13.1. 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.

PIECEWISE EXECUTION

As previously described, each code in the package is comprised of three major functionally independent modules: the Input Module, a 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 code package.

Module Execution Control

The execution of each of the three major modules in the 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 a previous table. Accordingly, if any Block-II through Block-VI card-image input is provided, the Input Module will be executed and the appropriate interface files created.

The execution of the Input Module will be suppressed if 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 the code.

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 and ensuring that there is no other SOLINP file present.

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 and ensuring that there is no other EDITIT file present.

Submodule Execution Control (File Generation Suppression)

The Input Module 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 file[1] from the Block-II card-image input data described in any of the User's Guides. This submodule will be executed and a GEODST file created by (i) setting (or defaulting) the Block-I input parameters 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."

2. Mixing Submodule Execution Control.

   The mixing submodule creates the standard interface files NDXSRF and ZNATDN[1] from the Block-IV card-image input data found in the MATLS array and, optionally, the PREMIX array as described in a User's Guide.

   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 (i) NOMIX is set to unity or (ii) the MATLS input array is omitted from the Block-IV card-image input or (iii) LIB= MACRXS or LIB= MACBCD in Block-III.

3. Assignment-of-Materials-to-Zones Submodule Execution Control.

   The assignment-of-materials-to-zones submodule 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 the User's Guides.

   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.

   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 or LIB= MACBCD.

   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 or LIB= MACBCD.

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 calculationally ordered 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 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 processes the Block-VI card-image input and creates the code-dependent interface file EDITIT for use by the EDIT module of DANTSYS.

   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 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 "Adjoint Computations" on page 7-37.

   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 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 is set to zero.

STACKED RUNS

It is possible to run more than one problem in a single execution of the code by stacking the problem-specification input "decks." In the context of this discussion the term "deck" refers to all card-image input necessary for a problem. To run more than one problem, the input file is created with two or more problem decks separated from one another by a single card-image record containing the entry

```
]eof
```

beginning in column 1. Thus, a single input file containing the specification decks for three separate problems would be constructed as follows:

```
Problem 1 "deck"
]eof
Problem 2 "deck"
]eof
Problem 3 "deck"
```

The code output for each of these problems will appear consecutively in a single output file.

Caution: Nonunique interface files created in one problem (for example, an RTFLUX scalar flux file) will be overwritten and lost when the next problem in the stack is executed.

Should a fatal error occur in a problem, the input for any remaining problems will be ignored.

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 Programming Language FORTRAN, ANSI X3.9-1978, American National Standards Institute, Inc., New York, NY 10018.

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, "An Adaptive Weighted Diamond-Differencing Method for Three-Dimensional XYZ Geometry." *Trans. Am. Nuc. Soc.* **68,** Part A (1993).

6. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," *Nucl. Sci. Eng.* **64**, 344 (1977).

ERROR MESSAGES

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

14

XTM — Transport IMethods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

INPUT ERRORS

A comprehensive error-checking capability has been provided in the DANTSYS 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.

Input errors will cause the run to terminate before entering the Solver Module. But one important feature of the error diagnostics in the Input Module is that an error will normally not cause an immediate termination. Instead, the code will attempt to process the remaining data in the offending input Block and/or in remaining input Blocks. Only after all remaining input has been processed (if possible) will 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.

Examples of Errors and Resulting Messages

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

Sample Error 1. 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:

```
*error* card       7 *2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2
                       xmessh=0.0  5.0  xints=5  zones=1    t

*error* array name xmessh   array number     -1

*error* columns  2 -  8


undefined array name
```

The first line of the error message indicates that an error was found on input card 7. 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 2 through 8 on the card-image. Finally, the message that the array name is undefined is provided.

This error is severe enough that further processing of the input is not attempted and no RUN HIGHLIGHTS are produced.

Sample Error 2. Input Block Terminator Omitted

As described in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1, 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. 2 0
* 2. Sample ONEDANT input to display error messages
* 3. Error 2 Terminal "t" omitted from Block I
* 4. /*********************** B L O C K I **********************
* 5. igeom=slab ngroup=30 isn=8 niso=16 mt=1 nzone=1 im=1 it=5
* 6. /*********************** B L O C K II *********************
* 7. xmesh=0.0 5.0 xints=5 zones=1 t
* 8. /*********************** B L O C K III ********************
* 9. lib= bxslib t
* 10. /*********************** B L O C K IV *********************
* 11. matls=matt 92235.50 assign=matls t
*
************************************************************************
```

Note that there is no delimited T at the end of line 5. As a result of this omission the following message is printed:

```
************************************************************************

***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 5 of the input but has continued reading until it found the terminal T following the three array entries belonging to Block-II on line 7 of the input.) The second line of the error message indicates that arrays that do not belong in Block-I have been found. 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 3. Invalid Entry for Block-I Input Parameter.

As described in any of the user's guides under "Block-I Details: Dimensions and Controls ," certain card-image input is always required in Block-I for a DANTSYS code execution. Specifically, the eight parameters IGEOM, NGROUP, ISN, NISO, MT, NZONE, IM, and IT are required to be entered as positive integers for ONEDANT. In this example, one of these parameters, IM, 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
**********************************************************************


* im
```

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

Sample Error 4. Incorrect Number of Entries in an Input Array.

Several input arrays available as input to DANTSYS 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 DANTSYS is shown below.

```
**********************************************************************

***error** fine mesh specs.ne.it
**********************************************************************
```

The error message is self-explanatory.

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

```
****************************************
*                                      *
*    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.   *
* *left boundary condition overridden* *
*     interface file solinp written.   *
*   edit module execution 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 5. Misplaced Array Identifier.

As discussed in the chapter "FREE FIELD INPUT REFERENCE" starting on page 9-1, 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.

The first line of the message indicates that an error was found on input card 12 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 32 of the card-image and the fourth line provides the self-explanatory message that a blank was found preceding the array identifier.

```
*error* card      12 *2 4 6 8(1)2 4 6 8(2)2 4 6 8(3)2 4 6 8(4)2 4 6
                     matls=matt 92235.50    assign =matls     t

*error* columns 32 - 32


**error**blank preceding an array identifier(=, $, or *)
```

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. 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

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

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.

IMPLEMENTATION ERRORS

A message from the code referring to an implementation error is intended to reflect a problem with the code implementation or coding and not to reflect an input error. This will normally require the intervention of a programmer. In some cases, it means there is a memory storage problem.

As previously mentioned, an implementation error should not be caused by an input error. However, some input errors can, after the error is detected and an error message printed, ultimately cause an implementation error message before aborting. So if the implementation error message is preceded by an input error message, the implementation error message may be ignored.

WARNINGS

Warning messages are less severe than error messages and will not cause the run to terminate. They do indicate that a check in the code has found something peculiar. It thus behooves the user to look at the provided information to determine if the code was performing as the user intended. An example of such a warning that can be caused by an anomaly in the input is when the user supplies explicitly a left boundary condition for a curvilinear geometry where the implicit left boundary is reflective. The warning message will indicate that the input value was overridden.

As with error messages, warning messages from the code will occur in two different places in the printed output. Some will appear as they are detected. Some will appear in the Run Highlights where they will be found surrounded by single asterisks. Double asterisks surround error messages in the Run Highlights. Thus, the more asterisks, the worse the condition.

During the course of the problem iteration, warning messages may appear in the iteration monitor. Although succinct, these are usually self explanatory. These may or may not be due to subtle input errors of some sort. A discussion of these for each of the solvers is found in "Iteration Monitor Print" on page 7-21.

FILE DESCRIPTIONS

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

15

XTM —Transport Methods Group

Los Alamos
National Laboratory

TABLE OF CONTENTS

INTRODUCTION

This chapter documents the detailed structure of all the files possibly used by the DANTSYS package. Included are input files, output files, and interface files, all of which are files that, once created, persist after the execution of the code is completed. Scratch files, which do not persist after execution, are not included.

Some of these files are ASCII text and some are binary sequential files. ASCII text files are used for the problem input file, some cross-section files, and some output files. Binary sequential files are used for all of the interface files and some of the input and output files.

File usage history, that is, where files are created and/or where they are read, is not documented here. See "CODE PACKAGE STRUCTURE" on page 13-9 for that.

ASCII FILES

Problem Input File

The input that describes the problem to be solved is code dependent and is the subject of the various Users' Guides in this document, which see. For instance, the TWODANT User's Guide describes the problem input for the TWODANT code.

Cross-Section Library Files

Cross sections may be provided in ASCII text form via the XSLIB, XSLIBB, or MACBCD files, or they may be embedded in the input file. See "COUPLED NEUTRON-GAMMA CROSS SECTIONS" on page 10-15 for a detailed description of these files.

EDTOUT

EDTOUT is a special ASCII file optionally prepared by the Edit Module of DANTSYS containing geometric and edit information which can be selectively processed by the user. This description of the EDTOUT file describes the format and construct of the file. The term "section" shall be used to refer to grouped data. The term "card image" or "card" shall be used in the same context that was described on page 9-13.

1. NUMBER-OF-TITLE-RECORDS (CARDS) SECTION (Format I6)

   This card contains the single number NTITLE, where NTITLE is the number of title records included in the file.

2. TITLE CARD SECTION (Format 10A8)

   The title cards from the problem input are given as individual records. This section is read as follows:

```
      DIMENSION HTITLE (10, NTITLE)
      DO 10 N=1,NTITLE
      READ (NINP,20) (HTITLE(I,N), I=1,10)
10    CONTINUE
20    FORMAT(10A8)
```

3. EDIT SPECIFICATION SECTION (Format 12I6)

   The edit specification section is a single card-image containing those parameters

needed to process the edit-related data sections. The entries in this section are ordered as follows:

a. IZNED Zone Edits present? 0/1=no/yes

b. NZNS Number of Edit Zones

c. IPTED Point Edits present? 0/1=no/yes

d. NIPES Number of points for point edits. -1/0/n = all points / no points / selected points

e. NEDISO Number of isotopes selected for edits. (Corresponds to EDISOS edits in DANTSYS)

f. MACRO Number of mixed macroscopic cross sections selected for edits. This corresponds to number of EDMATS entries PLUS the number of RESDNT edits in DANTSYS.

g. NCONS Number of constituent cross sections selected for edits. This corresponds to the number of EDCONS entries in DANTSYS

h. NXSTYP The number of cross-section types (positions) e.g. ABS, NUSIGF, in the edits. This corresponds to the number of EDXS entries in DANTSYS.

i. NSRF The number of response functions in the edits. This corresponds to the number of RSFNAM values in DANTSYS.

j. NBG The number of Edit Broad Groups in the edits.

k. ISADJ Adjoint problem? 0/1 = no/yes

4. GEOMETRIC SPECIFICATION SECTION (Format 12I6)

The geometric specification section contains those parameters needed to process the geometry-related data blocks. The 12 entries in this section are ordered as follows:

a. IDIMEN The geometry-dimension of the problem 1/2/3 = one-dimensional/two-dimensional/three-dimensional

b. IGEOM Geometry of the problem. 1=slab, 2=cylinder, etc.

c. IM Number of coarse radial mesh intervals

d. IT Total number of fine mesh intervals in radial direction

e. JM Number of coarse axial mesh intervals (=1 for IDIMEN>1)

f. JT Total number of axial fine mesh intervals (=1 for IDIMEN>1)

g. KM Number of coarse z mesh intervals (=1 for IDIMEN<3)

h. KT Number of fine z mesh intervals (=1 for IDIMEN<3)

i. NDUM1 Not used

j. NDUM2 Not used

k. NDUM3 Not used

l. NDUM4 Not used

ASIDE: In order to simplify the dimensions of some of the following data blocks, let us define the following internal parameters in terms of the preceding parameters found on the EDTOUT file:

a. NBGP1 = NBG + 1

b. NXSTOT = NEDISO + MACRO + NCONS. Note that an index I that runs from 1 to NXSTOT runs in the order indicated, namely isotopes, then mixed macroscopic, then constituents.

c. IMP1 = IM + 1

d. JMP1 = JM + 1

e. ITP1 = IT + 1

f. JTP1 = JT + 1

g. KMP1 = KM + 1

h. KTP1 = KT + 1

i. NIPE = a parameter which reflects the actual number of edit points. It assumes the following value as determined by the parameter NIPES:
NIPE = IT*JT*KT, if NIPES .LT. 0,
NIPE = 0, if NIPES .EQ. 0,
NIPE=NIPES, if NIPES .GT. 0

DATA BLOCKS

In reading the following data blocks, the user must assign data to be read to its own storage as defined by the parameters just defined. For the following data blocks we will give only a generic name for the data, the number of words in the data block, and the format-type of the block. REAL, INTEGER, and CHARACTER*8 data blocks are required. The presence or absence of a block will be indicated below by the IFF (if and only if) notation.

END ASIDE

5. GROUP ENERGY BOUNDS SECTION

ENERGY (NBGP1), 6E12

Using the data block ENERGY as an example, the block is to be read as follows:

```
DIMENSION ENERGY(NBGP1)
```

```
      READ (NINP,400) ENERGY
400   FORMAT(6E12.5)
```

6. POINTS EDITED SECTION [IFF (NIPES .GT. 0)]
   KPT (NIPE) , 12I6

7. COARSE RADIAL MESH BOUNDARIES SECTION
   XMESH (IMP1) , 6E12

8. NUMBER OF FINE RADIAL MESHES PER COARSE MESH SECTION
   IHX (IM) , 12I6

9. COARSE AXIAL MESH BOUNDARIES SECTION [IFF (IDIMEN .GT. 1)]
   YMESH (JMP1) , 6E12

10. NUMBER OF FINE AXIAL MESHES PER COARSE MESH SECTION
    [IFF (IDIMEN .GT. 1)]
    IHY (JM) , 12I6

11. COARSE Z MESH BOUNDARIES SECTION [IFF (IDIMEN .GT. 2)]
    ZMESH (KMP1) , 6E12

12. NUMBER OF FINE Z MESHES PER COARSE MESH SECTION
    [IFF (IDIMEN .GT. 2)]
    IHZ (KM) , 12I6

13. ZONE VOLUMES SECTION [IFF (NZNS .GT. 0)]
    VZ (NZNS) , 6E12

14. NAMES OF ISOTOPES SECTION [IFF (NEDISO .GT. 0)]
    HISO (NEDISO) , 9A8

15. NAMES OF MACROSCOPIC-EDITS SECTION [IFF (MACRO .GT. 0)]
    HMACR(MACRO) , 9A8

16. NAMES OF CONSTITUENTS SECTION [IFF (NCONS .GT. 0)
    HCONS (NCONS) , 9A8

17. NAMES OF CROSS SECTION EDIT-TYPES [IFF (NXSTYP .GT. 0)]
    HXSTY (NXSTYP) , 9A8

18. NAMES OF RESPONSE-FUNCTIONS SELECTED [IFF (NRSF .GT. 0)]

HRSF (NRSF) , 9A8

19. ZONES REACTION RATES SECTION [IFF (IZNED .GT. 0)]

A. CROSS-SECTION REACTION RATES SECTION [IFF (NXSTOT*NXSTYP .GT. 0)]

XSZN (NBGP1,NZNS,NXSTYP,NXSTOT) , 6E12

This section is to be read as follows:

```
      DIMENSION XSZN (NBGP1,NZNS,NXSTYP,NXSTOT)
      DO 40 L=1,NXSTOT
      DO 30 K=1,NXSTYP
      DO 20 J=1,NZNS
      READ (NINP,400) (XSZN(I,J,K,L), I=1,NBGP1)
 20   CONTINUE
 30   CONTINUE
 40   CONTINUE
400   FORMAT(6E12.5)
```

B. RESPONSE-FUNCTION REACTION RATES SECTION[IFF (NRSF .GT. 0)]

RSZN (NBGP1,NZNS,NRSF) , 6E12

This section is to be read as follows:

```
      DIMENSION RSZN (NBGP1,NZNS,NRSF)
      DO 30 K=1,NRSF
      DO 20 J=1,NZNS
      READ (NINP,400) (RSZN(I,J,K), I=1,NBGP1)
 20   CONTINUE
 30   CONTINUE
400   FORMAT(6E12.5)
```

20. POINT REACTION RATES SECTION [IFF (NIPE .GT. 0)]

A. CROSS-SECTION REACTION RATES SECTION [IFF (NXSTOT*NXSTYP .GT. 0)]

XSPT (NBGP1,NIPE,NXSTYP,NXSTOT) , 6E12

This section is to be read as follows:

```
      DIMENSION XSPT (NBGP1,NIPE,NXSTYP,NXSTOT)
      DO 40 L=1,NXSTOT
      DO 30 K=1,NXSTYP
```

```
      DO 20 J=1,NIPE
      READ (NINP,400) (XSPT(I,J,K,L), I=1,NBGP1)
 20   CONTINUE
 30   CONTINUE
 40   CONTINUE
400   FORMAT(6E12.5)
```

B. RESPONSE-FUNCTION REACTION RATES SECTION [IFF (NRSF .GT. 0)]

RSPT (NBGP1,NIPE,NRSF) , 6E12

This section is to be read as follows:

```
      DIMENSION RSPT (NBGP1,NIPE,NRSF)
      DO 30 K=1,NRSF
      DO 20 J=1,NIPE
      READ (NINP,400) (RSPT(I,J,K), I=1,NBGP1)
 20   CONTINUE
 30   CONTINUE
400   FORMAT(6E12.5)
```

EDTOGX

EDTOGX is a special ASCII file optionally prepared by the Edit Module of DANTSYS containing geometric, fission source, and scalar flux information which can be selectively processed by the user.

This description of the EDTOGX file describes the format and construct of the file. The term "section" shall be used to refer to grouped data. The term "card image" or "card" shall be used in the same context that was described on page 9-13.

1. NUMBER-OF-TITLE-RECORDS (CARDS) SECTION (Format I6)

   This card contains the single number NTITLE, where NTITLE is the number of title cards included in the file.

2. TITLE CARD SECTION (Format 10A8)

   The title cards from the problem are given as individual records. This section is read as follows

```
      DIMENSION HTITLE (10, NTITLE)
      DO 10 N=1,NTITLE
      READ (NINP,20) (HTITLE(I,N), I=1,10)
10    CONTINUE
20    FORMAT(10A8)
```

3. SPECIFICATION SECTION (Format 12I6)

   The specification section is a single card-image containing those parameters needed to process the data sections. The 12 entries in this section are ordered as follows:

   a. IDIMEN The geometry dimension of the problem 1/2/3 = one-dimensional/two-dimensional/three-dimensional

   b. ISADJ Adjoint problem? , 0/1 = no/yes

   c. NGROUP Number of energy groups

   d. IM Number of radial coarse mesh intervals

   e. IT Total number of radial fine mesh intervals.

   f. JM Number of axial coarse mesh intervals (=1 for IDIMEN<2)

   g. JT Total number of axial fine mesh intervals (=1 for IDIMEN<2)

   h. KM Number of z coarse mesh intervals (=1 for IDIMEN<3)

   i. KT Total number of z fine mesh intervals. (=1 for IDIMEN<3)

   j. IFISS Fission source array present? 0/1 = no/yes

   k. IGEOM Geometry of the problem. 1=slab, 2=cylinder, etc.

l. IADDFX Scalar fluxes present? 0/1 = yes/no

ASIDE: In order to simplify the dimensions of some of the following data blocks, the following four internal parameters are defined in terms of the preceding parameters found on the EDTOGX file:

a. IMP1 = IM + 1

b. JMP1 = JM + 1

c. KMP1 = KM + 1

d. IMJMKM = IM*JM*KM (Total number of coarse mesh intervals)

e. ITJTKT = IT*JT*KT (Total number of fine mesh intervals)

DATA BLOCKS

In reading the following data blocks, the user must assign data to be read to its own storage as defined by the parameters just defined. For the following data blocks we will give only a generic name for the data, the number of words in the data block, and the format-type of the block. REAL, INTEGER, and CHARACTER*8 data blocks are required. The presence or absence of a block will be indicated below by the IFF (if and only if) notation.

END ASIDE

4. RADIAL DATA INFORMATION

A. FINE MESH CELL-AVERAGE-RADIUS SECTION

RDAVG (IT) , 6E12

This section is to be read as follows:

```
      DIMENSION RDAVG (IT)
      READ (NINP,400) RDAVG
400   FORMAT(6E12.5)
```

B. NUMBER OF RADIAL-FINE-MESHES-PER-COARSE MESH SECTION

IHX (IM) , 12I6

C. COARSE MESH RADIAL BOUNDARIES SECTION

XMESH (IMP1) , 6E12

5. AXIAL DATA INFORMATION [IFF (IDIMEN .GT. 1)]

A. FINE MESH CELL-AVERAGE-AXIAL-POSITION SECTION

ADAVG (JT) , 6E12

B. NUMBER OF AXIAL-FINE-MESHES-PER-COARSE MESH SECTION

IHY (JM) , 12I6

C. COARSE-MESH-AXIAL-BOUNDARIES SECTION

YMESH (JMP1) , 6E12

6. Z DATA INFORMATION [IFF (IDIMEN .GT. 2)]

A. FINE MESH CELL-AVERAGE-Z-POSITION SECTION

ZDAVG (KT) , 6E12

B. NUMBER OF Z-FINE-MESHES-PER-COARSE MESH SECTION

IHZ (KM) , 12I6

C. COARSE-MESH-Z-BOUNDARIES SECTION

ZMESH (KMP1) , 6E12

7. ZONE NUMBERS-BY-COARSE-MESH-INTERVAL SECTION

IDCS (IMJMKM) , 12I6

8. FISSION-SOURCE-RATE SECTION [IFF (IFISS .GT. 0)]

FISRT (ITJTKT) , 6E12

9. SCALAR FLUX SECTION [IFF (IADDFX .EQ. 0)]

FLUX (ITJTKT,NGROUP) , 6E12

This section is to be read as follows:

```
DIMENSION FLUX (ITJTKT,NGROUP)
DO 10 J=1,NGROUP
```

```
      READ (NINP,400) (FLUX(I,J) , I=1,ITJTKT)
 10   CONTINUE
400   FORMAT(6E12.5)
```

ARBFLUX

The ARBFLUX file is a special ASCII file containing the angular fluxes for each fine-mesh boundary in a specified plane in the problem. This file is optionally output from TWODANT only and can be used to restart the problem as a boundary source calculation (see page 3-52). The ARBFLUX file consists of up to four subfiles, one each for ASRITE, ASBOTT, ASTOP, and ASLEFT. Each subfile is composed of two sections, where the Angular Flux Data Section is repeated NGROUP*JT times for the ASLEFT and ASRITE subfiles, and NGROUP*IT for the ASBOTT and ASTOP.

1. TITLE CARD SECTION (Format A72)

   This card identifies the direction of the angular flux (in-going for ASLEFT, out-going for ASRITE, down-going for ASBOTT, and up-going for ASTOP), and the plane it was written from.

2. ANGULAR FLUX DATA SECTION (Format 6(1x, 1pe11.5))

   This section contains mm*2 values of the angular boundary flux for a given cell and energy group. The ordering of the angles and quadrants is consistent with that required for inputting a full angular boundary source (see page 3-59).

   The string separator (Format 1x, ';') necessary for DANTSYS array input is written in a separate line. The string separator is output at the end of every Angular Flux Data Section in a subfile, except after the last entry.

STANDARD INTERFACE FILES

DANTSYS makes use of interface files to transmit data between and within its modules. These interface files are binary, sequential data files.

These files are of two types, standard or code-dependent. Standard interface files are interface files whose structure and data-content formats have been standardized by the Committee on Computer Code Coordination (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 DANTSYS: ISOTXS, GRUPXS, GEODST, NDXSRF, ZNATDN, SNCONS, FIXSRC, RTFLUX, ATFLUX, RAFLUX, AAFLUX and RZFLUX. File descriptions for these files are provided in Ref. 3.

CODE DEPENDENT INTERFACE FILES

DANTSYS makes use of interface files to transmit data between and within its modules. These interface files are binary, sequential data files.

These files are of two types, standard or code-dependent. Standard interface files are interface files whose structure and data-content formats have been standardized by the Committee on Computer Code Coordination (CCCC). Code-dependent interface files are files whose structure and data-content formats have not been standardized.

The following code-dependent binary interface files are used in all of the codes in the DANTSYS package: MACRXS, SNXEDT, ADJMAC, ASGMAT, SOLINP, and EDITIT.

Other code-dependent binary files provided solely as output from DANTSYS, but intended to serve as interfaces to other codes or as input to subsequent runs of the same code, are the UCFLUX, BXSLIB, RMFLUX/AMFLUX, RZMFLX, RAFLXM/ AAFLXM, and FISSRC files.

An extended GEODST, capable of describing the more complicated geometries used by the TWODANT/GQ code, is normally produced in a run of that code. The standard GEODST is a subset of this extended GEODST. This extended version, then, is a code dependent file and is listed here. TWODANT/GQ also writes a GEOSING file which is in extended GEODST format and is simply the GEODST file with a single submesh describing the calculational mesh domain. It is used solely to communicate with the EDIT module, although it is also useful for postprocessing.

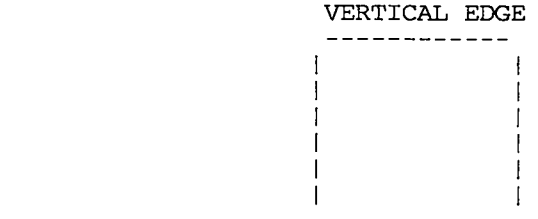
In this section are provided the file descriptions and a brief description of function for these code-dependent binary files. The file descriptions follow the format used for the standard interface file descriptions in Ref. 1 and Ref. 2.

AAFLXM for TWODANT

The AAFLXM file is a binary, code-dependent file containing the adjoint angular fluxes at each fine-mesh boundary. It differs from the standard adjoint angular flux file AAFLUX only in that the fluxes are in different angular order. In AAFLXM, the fluxes are in calculational order according to the sweeping; i.e., angles for each level of a quadrant.

```
C*************************************************************************
C                        REVISED 11/30/76                                -
C                                                                        -
CF          AAFLUX-IV                                                    -
CE          ADJOINT ANGULAR FLUX                                         -
C                                                                        -
C*************************************************************************

CD                   ORDER OF GROUPS IS ACCORDING TO INCREASING
CD                   ENERGY.  SINCE THE DIRECTION NUMBERS M=1,NDIR
CD                   ARE THOSE GIVEN IN THE SNCONS FILE, THE DATA
CD                   AS READ FROM THIS FILE IS FOR THE REFLECTED
CD                   DIRECTIONS.  NB: THE ORDER OF THE ANGLES IS
CD                   THE SAME AS THE SWEEP ORDER, BY LEVELS FOR
CD                   EACH QUADRANT.
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CS          FILE STRUCTURE                                               -
CS                                                                       -
CS             RECORD TYPE                          PRESENT IF           -
CS             ============================         ==========           -
CS             FILE IDENTIFICATION                  ALWAYS               -
CS             FILE CONTROL                         ALWAYS               -
CS                                                                       -
CS   ************(REPEAT FOR ALL GROUPS)                                -
CS   *               (GROUP 1 IS FIRST)                                  -
CS   *   *********(REPEAT FOR ALL DIRECTIONS)                            -
CS   *   *     COMPUTATIONAL COSINES                ALWAYS               -
CS   *   *     HORIZONTAL-EDGE ANGULAR FLUX         ALWAYS               -
CS   *   *     VERTICAL-EDGE ANGULAR FLUX           ALWAYS               -
CS   *   *********                                                       -
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 - AAFLUX - (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-----------------------------------------------------------------------
CR          SPECIFICATIONS      (1D RECORD)                             -
C                                                                       -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,NDIR,EFFK,ADUM                      -
C                                                                       -
CW    8=NUMBER OF WORDS                                                 -
C                                                                       -
CD    NDIM              NUMBER OF DIMENSIONS                            -
CD    NGROUP            NUMBER OF GROUPS                                -
CD    NINTI             NUMBER OF FIRST DIMENSION FINE MESH INTERVALS   -
CD    NINTJ             NUMBER OF SECOND DIMENSION FINE MESH INTERVALS.-
CD                      NINTJ.EQ.1 IF NDIM.EQ.1                         -
CD    NINTK             NUMBER OF THIRD DIMENSION FINE MESH INTERVALS.  -
CD                      NINTK.EQ.1 IF NDIM.LE.2                         -
CD    NDIR              NUMBER OF DIRECTIONS                            -
CD    EFFK              EFFECTIVE MULTIPLICATION FACTOR                 -
CD    ADUM              RESERVED                                        -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR          HORIZONTAL EDGE ADJOINT ANGULAR FLUX                        -
C                                                                       -
CL    ((HEDGE(I,J),I=NBDRYI),J=1,NINTJ)                                 -
C                                                                       -
CW    NBDRYI*NINTJ*MULTI=NUMBER OF WORDS                                -
C                                                                       -
CD    HEDGE(I,J)    HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX               -
C                                                                       -
CD    NBDRYI        NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH        -
CD                  BOUNDARIES)                                         -
C                                                                       -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR          VERTICAL EDGE ADJOINT ANGULAR FLUX                          -
C                                                                       -
CL    ((VEDGE(I,J),I=NINTI),J=1,NBDRYJ)                                 -
C                                                                       -
CW    NINTI*NBDRYJ*MULTI=NUMBER OF WORDS                                -
C                                                                       -
CD    VEDGE(I,J)    HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX               -
C                                                                       -
CD    NBDRYJ        NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH       -
CD                  BOUNDARIES)                                         -
C                                                                       -
C                                                                       -
C                                                                       -
C-----------------------------------------------------------------------
```




```
CEOF
```

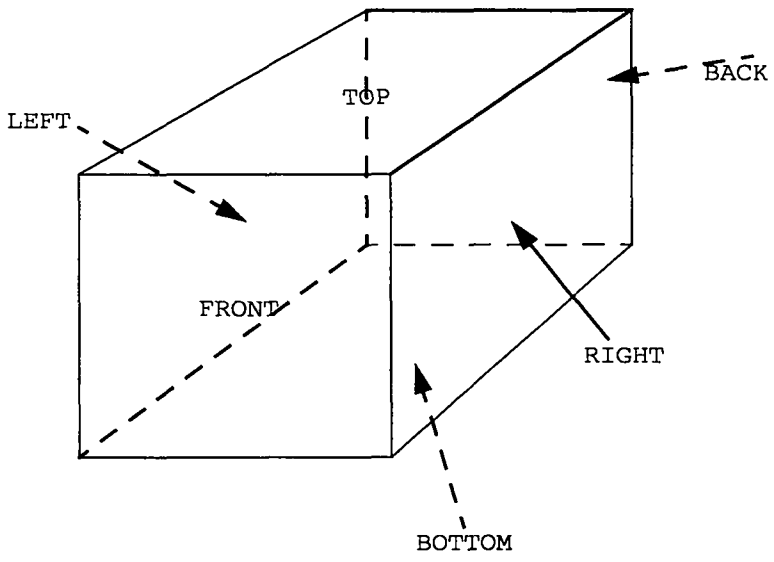
AAFLXM for THREEDANT

The AAFLXM file is a binary, code-dependent file containing the angular fluxes for the adjoint flux at each fine-mesh boundary. It differs from the standard adjoint angular flux file AAFLUX only in that the fluxes are in different order. In AAFLXM, the fluxes are in calculational order.

```
C*********************************************************************-
C                          DATE 3/03/95                                -
C                                                                      -
CF          AAFLXM                                                     -
CE          CODE DEPENDENT COMPUTATIONAL-ORDERED ADJOINT ANGULAR FLUX  -
CE          AT CELL EDGES FOR THREEDANT CODE                           -
C                                                                      -
C*********************************************************************-


C---------------------------------------------------------------------
C                                                                      -
C                                                                      -
CN               THIS FILE PROVIDES THE EDGE ANGULAR FLUX AS ORDERED   -
CN               BY THE CODE                                           -
C                                                                      -
C                                                                      -
C---------------------------------------------------------------------


C---------------------------------------------------------------------
C                                                                      -
CD                      ORDER OF GROUPS IS ACCORDING TO INCREASING     -
CD                      ENERGY. NOTE THAT DOUBLE PRECISION FLUXES ARE  -
CD                      GIVEN WHEN MULT=2                              -
C                                                                      -
C---------------------------------------------------------------------


C---------------------------------------------------------------------
CS          FILE STRUCTURE                                             -
CS                                                                     -
CS             RECORD TYPE                          PRESENT IF         -
CS             ============================         ==========         -
CS             FILE IDENTIFICATION                  ALWAYS             -
CS             FILE CONTROL                         ALWAYS             -
CS                                                                     -
CS   *************(REPEAT FOR ALL GROUPS)                            -
CS   *                 (GROUP NGROUP IS FIRST)                        -
CS   *     ********(REPEAT FOR ALL FRONT-GOING DIRECTIONS)            -
CS   *     *   COMPUTATIONAL COSINES                 ALWAYS           -
CS   *     *   CELL BACK-EDGE ANGULAR FLUX FOR BACK PLANE             -
CS   *     *******                                                    -
CS   *                                                                -
CS   *     *******(REPEAT FOR ALL Z-INTERVALS)                       -
CS   *                 (INTERVAL NINTK IS FIRST)                      -
CS   *     *  ******(REPEAT FOR ALL FRONT-GOING DIRECTIONS)           -
CS   *     *  * COMPUTATIONAL COSINES               ALWAYS           -
CS   *     *  * HORIZONTAL-EDGE ANGULAR FLUX        ALWAYS           -
CS   *     *  * VERTICAL-EDGE ANGULAR FLUX          ALWAYS           -
CS   *     *  * CELL FRONT-EDGE ANGULAR FLUX        ALWAYS           -
CS   *     *  *****                                                   -
CS   *     * ******                                                   -
CS   *                                                                -
CS   *     ********(REPEAT FOR ALL BACK-GOING DIRECTIONS)            -
```

```
CS    *     *   COMPUTATIONAL COSINES                ALWAYS                -
CS    *     *   CELL FRONT-EDGE ANGULAR FLUX FOR FRONT PLANE              -
CS    *     *******                                                       -
CS    *                                                                   -
CS    *     *******(REPEAT FOR ALL Z-INTERVALS)                           -
CS    *                 (INTERVAL 1 IS FIRST)                             -
CS    *     *  ******(REPEAT FOR ALL BACK-GOING DIRECTIONS)               -
CS    *     *  * COMPUTATIONAL COSINES              ALWAYS                -
CS    *     *  * HORIZONTAL-EDGE ANGULAR FLUX       ALWAYS                -
CS    *     *  * VERTICAL-EDGE ANGULAR FLUX         ALWAYS                -
CS    *     *  * CELL BACK-EDGE ANGULAR FLUX        ALWAYS                -
CS    *     *  *****                                                      -
CS    *     * ******                                                      -
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 - RAFLXM - (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-------------------------------------------------------------------------
CR          SPECIFICATIONS      (1D RECORD)                               -
C                                                                         -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,EFFK,POWER                            -
C                                                                         -
CW    8 =NUMBER OF WORDS                                                  -
C                                                                         -
CD    NDIM               NUMBER OF DIMENSIONS                           -
CD    NGROUP             NUMBER OF ENERGY GROUPS                        -
CD    NINTI              NUMBER OF FIRST DIMENSION FINE MESH INTERVALS  -
CD    NINTJ              NUMBER OF SECOND DIMENSION FINE MESH INTERVALS -
CD    NINTK              NUMBER OF THIRD DIMENSION FINE MESH INTERVALS. -
CD                       NINTK.EQ.1 IF NDIM.LE.2                        -
CD    EFFK               EFFECTIVE MULTIPLICATION FACTOR                -
CD    POWER              POWER IN WATTS TO WHICH FLUX IS NORMALIZED      -
C                                                                         -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR          COMPUTATIONAL COSINES                                         -
C                                                                         -
CL    XMU,XETA,XI,WEIGHT,M,K                                              -
C                                                                         -
CW    6*MULT                                                              -
C                                                                         -
CD    XMU            MU COSINE                                          -
CD    XETA           ETA COSINE                                         -
CD    XI             XI COSINE                                          -
CD    WEIGHT         WEIGHT                                             -
```

```
CD    M              ANGLE INDEX IN THE OCTANT                          -
CD    K              Z-PLANE NUMBER                                     -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR           CELL FRONT-EDGE ANGULAR FLUX                               -
C                                                                       -
CL    ((FBEDGE(I,J),I=NINTI),J=1,NINTJ)                                 -
C                                                                       -
CW    NINTI*NINTJ*MULT=NUMBER OF WORDS                                  -
C                                                                       -
CD    FBEDGE(I,J)     CELL FRONT-EDGE ANGULAR FLUX                      -
C                                                                       -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR           HORIZONTAL-EDGE ANGULAR FLUX                               -
C                                                                       -
CL    ((HEDGE(I,J),I=1,NBDRYI),J=1,NINTJ)                               -
C                                                                       -
CD    HEDGE(I,J)     HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX              -
C                                                                       -
CW    NBDRYI*NINTJ*MULT=NUMBER OF WORDS                                 -
C                                                                       -
CD    NBDRYI         NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH       -
CD                   BOUNDARIES)                                        -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR           VERTICAL-EDGE ANGULAR FLUX                                 -
C                                                                       -
CL    ((VEDGE(I,J),I=1,NINTI),J=1,NBDRYJ)                               -
C                                                                       -
CD    VEDGE(I,J)     VERTICAL-EDGE-BOUNDARY ANGULAR FLUX                -
C                                                                       -
CW    NINTI*NBRDYJ*MULT=NUMBER OF WORDS                                 -
C                                                                       -
CD    NBDRYJ         NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH      -
CD                   BOUNDARIES)                                        -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR           CELL BACK-EDGE ANGULAR FLUX                                -
C                                                                       -
CL    ((FBEDGE(I,J),I=NINTI),J=1,NINTJ)                                 -
C                                                                       -
CW    NINTI*NINTJ*MULT=NUMBER OF WORDS                                  -
C                                                                       -
CD    FBEDGE(I,J)     CELL BACK-EDGE ANGULAR FLUX                       -
C                                                                       -
C-----------------------------------------------------------------------
```



CEOF

ADJMAC

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

```
C*****************************************************************-
C                          DATE 05/12/83                          -
C                                                                 -
CF          ADJMAC                                                -
CE          CODE DEPENDENT MACROSCOPIC MULTIGROUP CROSS SECTION FILE -
CE          USED IN ONEDANT SOLVER MODULE FOR ADJOINT CALCULATIONS -
C                                                                 -
C                                                                 -
C*****************************************************************-
C                                                                 -
CN                THIS FILE PROVIDES A BASIC BROAD GROUP          -
CN                LIBRARY, ORDERED BY GROUP                       -
C                                                                 -
CN                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.0     -
CS   *         SCATTERING MATRIX                    NORD.NE.0     -
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 - ADJMAC - (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-----------------------------------------------------------------
CR          FILE CONTROL                                          -
C                                                                 -
CL    NGROUP,NMAT,NORD,NED,IDPF,LNG,MAXUP,MAXDN,NPRIN,I2LP1       -
C                                                                 -
CW    10=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 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                    -
CD    NPRIN         NUMBER OF PRINCIPAL CROSS SECTIONS                      -
CD    I2LP1         0/1 = NO/YES  2L+1 TERM WAS INCLUDED IN LIBRARY         -
C                                                                           -
C-------------------------------------------------------------------------
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)*MULT=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 BOUND OF GROUP N (EV)                    -
CD    EMIN          MINIMUM ENERGY BOUND OF SET (EV)                        -
CD    NEDT          NED+NPRIN                                               -
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                                                                           -
CN                  ALSO PRESENT WHEN NPRIN=5 IS:                           -
C                                                                           -
CN                        HED(5) = 5HTRANS                                  -
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*MULT=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                                                                           -
CN                  ALSO PRESENT WHEN NPRIN=5 IS:                           -
C                                                                           -
CN                       J=5  TRANSPORT CROSS SECTION                       -
C                                                                           -
C-------------------------------------------------------------------------
C                                                                           -
C-------------------------------------------------------------------------
CR          SCATTERING CONTROL BLOCK FOR GROUP N                            -
```

```
C                                                                       -
CC         PRESENT IF NORD.GT.0                                         -
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.0                                         -
C                                                                       -
CL     (SCAT(I),I=1,NTAB)                                               -
C                                                                       -
CW     NTAB*MULT=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
```

AMFLUX

The AMFLUX code-dependent file contains, in binary form, the spherical harmonics adjoint angular flux moments for all spatial fine mesh points and all energy groups. It is optionally produced by the Solver Module.

```
C*************************************************************************
C                        DATE 04/01/85                                    -
C                                                                         -
CF          AMFLUX-IV                                                     -
CE          ADJOINT FLUX MOMENTS                                          -
C                                                                         -
C*************************************************************************
C                                                                         -
CN              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             SPECIFICATIONS                       ALWAYS                -
CS                                                                        -
CS   ************(REPEAT FOR ALL GROUPS)                                 -
CS   *         ADJOINT MOMENTS FLUXES                ALWAYS                -
CS   *************                                                       -
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 - RMFLUX - (A6)              -
CD    HNAME              HOLLERITH FILE NAME -        -(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          SPECIFICATIONS     (1D RECORD)                                -
C                                                                         -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,NORD,EFFK,POWER,OITNO                 -
C                                                                         -
CW    9=NUMBER OF WORDS                                                   -
C                                                                         -
CD    NDIM               NUMBER OF DIMENSIONS                             -
CD    NGROUP             NUMBER OF GROUPS                                 -
CD    NINTI              NUMBER OF FIRST DIMENSION INTERVALS              -
CD    NINTJ              NUMBER OF SECOND DIMENSION INTERVALS             -
CD                       NINTJ.EQ.1 IF NDIM.EQ.1                          -
CD    NINTK              NUMBER OF THIRD DIMENSION INTERVALS              -
```

```
CD                        NINTK.EQ.1 IF NDIM.LE.2                          -
CD     NORD               NUMBER OF LEGENDRE MOMENTS                       -
CD     EFFK               EFFECTIVE MULTIPLICATION FACTOR                  -
CD     POWER              POWER IN WATTS TO WHICH FLUX IS NORMALIZED        -
CD     OITNO              OUTER ITERATION NUMBER                           -
C-------------------------------------------------------------------------
C                                                                           -
C-------------------------------------------------------------------------
CR          ADJOINT MOMENTS FLUXES ON MULTIDIMENSIONAL INTERVALS            -
C                              (2D RECORD)                                  -
C                                                                           -
CL     ((FLUX(M,I),M=1,NORD),I=1,NINTI)  ----NOTE STRUCTURE BELOW---        -
C                                                                           -
CW     NORD*NINTI=NUMBER OF WORDS                                           -
C                                                                           -
C      DO 1 K=1,NINTK                                                       -
C      DO 1 J=1,NINTJ                                                       -
C    1 READ (N) *LIST AS ABOVE*                                             -
C                                                                           -
CD     FLUX(M,I)          ADJOINT FLUX MOMENTS ON FIRST DIMENSION          -
CD                        INTERVALS.                                       -
C                                                                           -
C-------------------------------------------------------------------------
CEOF                                                                        -
```

ASGMAT

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
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.0
CS             MATERIAL LIST FOR ALL ZONES          MPZTOT.NE.0
CS             MATERIAL CONCENTRATIONS              MPZTOT.NE.0
CS             MATERIAL CONCENTRATION FACTORS       MPZTOT.NE.0
CS             CONCENTRATION MODIFIER               MPZTOT.NE.0
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 - ASGMAT - (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---------------------------------------------------------------------
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
CD   4  FMMIX     0/1 NO/YES VOL FRACTION MIXING BY FINE MESH
```

```
C                                                                               -
C-------------------------------------------------------------------------------
C                                                                               -
C-------------------------------------------------------------------------------
CR          COMPATABILITY CODE WORDS                                            -
C                                                                               -
CL    CODE1, CODE2                                                              -
C                                                                               -
CW    2*MULT=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*MULT=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*MULT=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.0                                                   -
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                                                                               -
C-------------------------------------------------------------------------------
CR          MATERIAL LIST FOR ALL ZONES                                         -
C                                                                               -
CC     PRESENT IF MPZTOT.NE.0                                                   -
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                                                                               -
C-------------------------------------------------------------------------------
```

```
CR          MATERIAL CONCENTRATIONS                                          -
C                                                                            -
CC     PRESENT IF MPZTOT.NE.0                                                -
C                                                                            -
CL    (CONC(I),I=1,MPZTOT)                                                   -
C                                                                            -
CW    MPZTOT*MULT=NUMBER OF WORDS                                            -
C                                                                            -
CD      CONC(I)     CONCENTRATION OF THE I-TH MATERIAL IN THE MATLST          -
CD                  ARRAY                                                    -
C                                                                            -
C-------------------------------------------------------------------------   -
C                                                                            -
C-------------------------------------------------------------------------   -
CR          MATERIAL CONCENTRATION FACTORS                                   -
C                                                                            -
CC     PRESENT IF MPZTOT.NE.0                                                -
C                                                                            -
CL    (C1(I),I=1,MPZTOT)                                                     -
C                                                                            -
CW    MPZTOT*MULT=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.0                                                -
C                                                                            -
CL    CMOD                                                                   -
C                                                                            -
CW    1*MULT=NUMBER OF WORDS                                                 -
C                                                                            -
CD      CMOD      INPUT VALUE OF THE CONCENTRATION MODIFIER                  -
C                                                                            -
C-------------------------------------------------------------------------   -
CEOF                                                                         -
```

AZMFLX

The AZMFLX file is a binary, code-dependent file containing the spherical harmonics adjoint angular flux moments averaged over each zone for each energy group. The zones over which the fluxes are averaged are the zones used in the Solver Module and not the Edit Zones optionally used in the Edit Module.

```
C*****************************************************************
C                        DATE 01/28/95                           -
C                                                                -
CF          AZMFLX-IV                                           -
CE          ADJOINT ZONE AVERAGED FLUX MOMENTS BY GROUP         -
C                                                                -
C*****************************************************************
C                                                                -
CN                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             SPECIFICATIONS                       ALWAYS      -
CS                                                              -
CS   *************(REPEAT FOR ALL GROUPS)                      -
CS   *         ZONE AVERAGED FLUX MOMENTS          ALWAYS      -
CS   *************                                             -
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 - RZMFLX - (A6)     -
CD    HNAME             HOLLERITH FILE NAME -        -(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          SPECIFICATIONS     (1D RECORD)                      -
C                                                               -
CL    NDIM,NGROUP,NZONE,DUM,DUM,NORD,EFFK,POWER,OITNO           -
C                                                               -
CW    9=NUMBER OF WORDS                                         -
C                                                               -
CD    NDIM              NUMBER OF DIMENSIONS                    -
CD    NGROUP            NUMBER OF GROUPS                        -
CD    NZONE             NUMBER OF GEOMETRIC ZONES               -
CD    DUM               DUMMY, NOT USED                         -
CD    DUM               DUMMY, NOT USED                         -
```

```
CD     NORD                NUMBER OF LEGENDRE MOMENTS                          -
CD     EFFK                EFFECTIVE MULTIPLICATION FACTOR                     -
CD     POWER               POWER IN WATTS TO WHICH FLUX IS NORMALIZED           -
CD     OITNO               OUTER ITERATION NUMBER                              -
C                                                                              -
C--------------------------------------------------------------------------------
C                                                                              -
C--------------------------------------------------------------------------------
CR           ADJOINT FLUX MOMENTS AVERAGED OVER EACH ZONE                      -
C                              (2D RECORD)                                     -
C                                                                              -
CL     ((FLUX(M,I),M=1,NORD),I=1,NZONE)                                        -
C                                                                              -
CW     NORD*NZONE=NUMBER OF WORDS                                              -
C                                                                              -
C                                                                              -
CD     FLUX(M,I)           ADJOINT FLUX MOMENT AVERAGES FOR EACH               -
CD                         ZONE.                                               -
C                                                                              -
C--------------------------------------------------------------------------------
CEOF                                                                           -
```

BXSLIB

The BXSLIB code-dependent file contains, in binary form, the cross sections and other cross section and mixing information as described in "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 10-12.

```
C*****************************************************************-
C                        DATE 09/22/88                             -
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. -
CN                       THE FILE ALSO MAY CONTAIN ISOTOPE/ATOMIC WEIGHT-
CN                       PAIRS FROM THE BLOCK IV ONEDANT INPUT     -
C                                                                  -
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 ISOTOPES)                        -
CS   *  **********(REPEAT FOR ALL LEGENDRE ORDERS)                 -
CS   *  *       CROSS SECTION SET                    ALWAYS         -
CS   *  **********                                                 -
CS   *************                                                -
CS             ISOTOPE LABEL/ATOMIC WEIGHT PAIR     NISOAW.GT.0    -
C                                                                  -
C-----------------------------------------------------------------
C-----------------------------------------------------------------
CR          FILE IDENTIFICATION                                    -
C                                                                  -
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(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-----------------------------------------------------------------
CR          FILE CONTROL                                           -
C                                                                  -
CL    NGROUP,NISO,IHM,IHT,IHS,MAXT,NISOAW,NXSREC,I2LP1             -
C                                                                  -
CW    9=NUMBER OF WORDS                                            -
C                                                                  -
C                                                                  -
```

```
CD    NGROUP        NUMBER OF ENERGY GROUPS IN FILE                          -
CD    NISO          NUMBER OF ISOTOPES IN FILE                               -
CD    IHM           TABLE LENGTH (NUMBER OF CROSS SECTIONS FOR               -
CD                  ONE ISOTOPE, FOR ONE GROUP, AND FOR ONE LEGENDRE         -
CD                  ORDER                                                    -
CD    IHT           TOTAL CROSS SECTION POSITION IN THE TABLE                -
CD    IHS           SELF SCATTER CROSS SECTION POSITION                      -
CD    MAXT          MAXIMUM NUMBER OF LEGENDRE MOMENTS                       -
CD    NISOAW        NUMBER OF ISOTOPE LABEL/ATOMIC WEIGHT PAIR               -
CD    NXSREC        TOTAL NUMBER OF CROSS SECTION SETS                       -
CD    I2LP1         0/1 - NO/YES  SCATTERING CROSS SECTIONS CONTAIN          -
CD                                2L+1 FACTOR                              -
C                                                                            -
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
CR          FILE DATA                                                        -
C                                                                            -
CL    (HSETID(I),I=1,12),(HISONM(I),I=1,NISO),(EDNAME(I),I=1,IHT-3),         -
CL   1(CHI(J),J=1,NGROUP),(VEL(J),J=1,NGROUP),                               -
CL   2(EMAX(J),J=1,NGROUP),EMIN,(NSPI(I),I=1,NISO)                           -
C                                                                            -
CW    (NISO+12+IHT-3)*MULT+(3*NGROUP+1)*MULT+NISO=NUMBER OF WORDS            -
C                                                                            -
C                                                                            -
CD    HSETID(I)     HOLLERITH IDENTIFICATION OF FILE (A6)                    -
CD    HISONM(I)     HOLLERITH ISOTOPE LABEL FOR ISOTOPE I (A6)               -
CD    EDNAME(I)     HOLLERITH NAME FOR EDIT POSITIONS                        -
CD                  PRECEDING SIGMA ABSORPTION FOR POSITION I (A6)           -
CD    CHI(J)        FILE-WIDE FISSION SPECTRUM(ZEROES IN ONEDANT)            -
CD    VEL(J)        MEAN NEUTRON VELOCITY IN GROUP J (CM/SEC)                -
CD    EMAX(J)       MAXIMUM ENERGY BOUND OF GROUP J (EV)                     -
CD    EMIN          MINIMUM ENERGY BOUND OF SET (EV)                         -
CD    NSPI(I)       NUMBER OF LEGENDRE ORDERS FOR ISOTOPE I                  -
C                                                                            -
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
CR          CROSS SECTION SET FOR ISOTOPE I AND LEGENDRE ORDER M             -
C                                                                            -
CL    ((C(I,J),I=1,IHM),J=1,NGROUP)                                          -
C                                                                            -
CW    IHM*NGROUP*MULT=NUMBER OF WORDS                                        -
C                                                                            -
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
CR          HOLLERITH ISOTOPE LABEL/ATOMIC WEIGHT PAIR SET                   -
C                                                                            -
CC   PRESENT IF NISOAW.GT.0                                                  -
C                                                                            -
CL    (ATWTP(I),I=1,2*NISOAW)                                                -
C                                                                            -
CW    2*NISOAW*MULT=NUMBER OF WORDS                                          -
C                                                                            -
C-------------------------------------------------------------------------------
CEOF                                                                         -
```

EDITIT

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 01/28/95                          -
C                                                                 -
CF          EDITIT                                                -
CE          CODE DEPENDENT FILE OF INFORMATION SPECIFIC TO THE    -
CE          ONEDANT EDIT MODULE                                   -
C                                                                 -
C                                                                 -
C*****************************************************************-
C                                                                 -
CN                THIS FILE CONTAINS THE CARD INPUT INFORMATION   -
CN                FROM BLOCK VI                                   -
C                                                                 -
C-----------------------------------------------------------------
CS      FILE STRUCTURE                                            -
CS                                                                -
CS         RECORD TYPE                              PRESENT IF    -
CS         ================================         ==========    -
CS         FILE IDENTIFICATION                      ALWAYS        -
CS         RAW CONTROLS AND DIMENSIONS              ALWAYS        -
CS         DEFAULTED CONTROLS AND DIMENSIONS        ALWAYS        -
CS         RAW FLOATING INPUT DATA                  ALWAYS        -
CS         FINE GROUPS PER BROAD GROUP              ALWAYS        -
CS         ZONE NUMBERS                             NZNS.NE.0     -
CS         POINT NUMBERS                            NIPE.NE.0     -
CS         IPLANE NUMBERS                           NIPLNE.NE.0   -
CS         JPLANE NUMBERS                           NJPLNE.NE.0   -
CS         KPLANE NUMBERS                           NKPLNE.NE.0   -
CS         CROSS SECTION POSITIONS                  NPOS.NE.0     -
CS         ISOTOPE NUMBERS TO EDIT                  NISO.NE.0     -
CS         MATERIAL NUMBERS TO EDIT                 MACRO.NE.0    -
CS         CONSTITUENT NUMBERS TO EDIT              NCONS.NE.0    -
CS         RESPONSE FUNCTION NAMES                  IDOSE.NE.0    -
CS   *************(REPEAT FOR ALL RESPONSE FUNCTIONS)             -
CS   *     RESPONSE FUNCTION ENERGY VECTOR          IDOSE.NE.0    -
CS   *     RESPONSE FUNCTION X SPATIAL VECTOR       IDOSE.NE.0    -
CS   *     RESPONSE FUNCT Y SPATIAL VECTOR  IDOSE.NE.0.AND.IDIMEN.GT.1 -
CS   *     RESPONSE FUNCT Z SPATIAL VECTOR  IDOSE.NE.0.AND.IDIMEN.GT.2 -
CS   *************                                               -
CS         CROSS SECTION SUMMING ARRAY -            IXSUM.NE.0    -
CS         RESPONSE FUNCTION SUMMING ARRAY          IRSUM.NE.0    -
CS         FINE X MESH DENSITY FACTORS              IDEN.NE.0     -
CS         FINE Y MESH DENSITY FACTORS     IDEN.NE.0 .AND. IDIMEN.GT.1 -
CS         FINE Z MESH DENSITY FACTORS     IDEN.NE.0 .AND. IDIMEN.GT.2 -
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 - 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 DOUBLE PRECISION WORD                 -
C                                                                            -
C-------------------------------------------------------------------------
C                                                                            -
C-------------------------------------------------------------------------
CR          RAW CONTROLS AND DIMENSIONS                                      -
C                                                                            -
CL    IEDOPT,PTED,  NIPE,  IKND,   ZNED, NZNS,IXSUM,IRSUM, NPOS, NISO,      -
CL  1 NCONS, MACRO, IDOSE, IGRPED, LNG,  NBG, IDEN, NGROUP, AJED,            -
CL  2 ITED, JTED, KTED, IRZFLX, IRZMFX, EDOUTF, IFLUX1, IPRPLT,              -
CL  3 NIPLNE, NJPLNE, NKPLNE                                                 -
C                                                                            -
CW    75=NUMBER OF WORDS                                                     -
C                                                                            -
CD   1  IEDOPT   NOT USED                                                    -
CD   2  PTED     0/1 - NO/YES  DO POINT EDIT                                 -
CD   3  NIPE     NUMBER OF POINTS TO EDIT                                    -
CD   4  BYVOLP   0/1 - NO/YES  MULTIPLY REACTION RATES BY MESH VOLUME        -
CD   5  ZNED     0/1 - 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                                  -
CD                                                                           -
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   0/1/2/3 - ENERGY GROUP PRINT OPTIONS                        -
CD  15  LNG      NUMBER OF THE LAST NEUTRON GROUP                            -
CD  16  NBG      NUMBER OF BROAD ENERGY GROUPS                               -
CD  17  IDEN     0/1 - NO/YES  THERE ARE FINE MESH DENSITY FACTORS            -
CD  18  NGROUP   NUMBER OF FINE ENERGY GROUPS                                -
CD  19  AJED     0/1 - NO/YES  THIS IS AN ADJOINT EDIT                       -
CD  20  ITED     NUMBER OF FINE RADIAL MESH                                  -
CD  21  JTED     NUMBER OF FINE AXIAL MESH                                   -
CD  22  KTED     NUMBER OF FINE Z DIRECTION MESH                             -
CD  23  IRZFLX   0/1 - NO/YES  WRITE CCCC RZFLUX FILE                        -
CD  24  IRZMFX   0/1 - NO/YES  WRITE ZONE MOMENTS FILE                       -
CD  25  EDOUTF   -3/-2/0/1/2/3 ASC EDIT FILE PREPARATION INDICATOR           -
CD  26  IFLUX1   0/1 - NO/YES  MAKE ALL INPUT FLUXES UNITY                   -
CD  27  IPRPLT   0/1/2/3 PRINT/NOTHING/TECPLOT/BOTH VISUALIZATION OPT        -
CD  28  NIPLNE   NUMBER OF I PLANES PLOTTED                                  -
CD  29  NJPLNE   NUMBER OF J PLANES PLOTTED                                  -
CD  30  NKPLNE   NUMBER OF K PLANES PLOTTED                                  -
C                                                                            -
C-------------------------------------------------------------------------
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-------------------------------------------------------------------------
C                                                                            -
C-------------------------------------------------------------------------
CR          RAW FLOATING INPUT DATA                                          -
C                                                                            -
CL    POWER, MEVPER                                                          -
C                                                                            -
```

```
CW    2*MULT=NUMBER OF WORDS                                                -
C                                                                           -
CD   1  POWER    NORMALIZE TO POWER                                         -
CD   2  MEVPER   MEV RELEASED PER FISSION                                   -
C                                                                           -
C-------------------------------------------------------------------------
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-------------------------------------------------------------------------
C                                                                           -
C-------------------------------------------------------------------------
CR          ZONE NUMBERS                                                    -
C                                                                           -
CC     PRESENT IF NZNS.GT.0                                                 -
C                                                                           -
CL    (EDZONE(I),I=1,IT*JT*KT)                                              -
C                                                                           -
CW    IT*JT*KT=NUMBER OF WORDS                                              -
C                                                                           -
CD      EDZONE(I) EDIT ZONE NUMBER FOR THE I-TH FINE MESH                   -
C                                                                           -
C-------------------------------------------------------------------------
C                                                                           -
C-------------------------------------------------------------------------
CR          POINTS TO EDIT                                                  -
C                                                                           -
CC     PRESENT IF NIPE.GT.0                                                 -
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          IPLANES TO EDIT                                                 -
C                                                                           -
CC     PRESENT IF NIPLNE.GT.0                                               -
C                                                                           -
CL    (IPLNES(I),I=1,NIPLNE)                                                -
C                                                                           -
CW    NIPLNE=NUMBER OF WORDS                                                -
C                                                                           -
CD      IPLNES(I) NUMBER OF THE I-TH PLANE TO EDIT      -                   -
C                                                                           -
C                                                                           -
C-------------------------------------------------------------------------
CR          JPLANES TO EDIT                                                 -
C                                                                           -
CC     PRESENT IF NJPLNE.GT.0                                               -
C                                                                           -
CL    (JPLNES(I),I=1,NJPLNE)                                                -
C                                                                           -
CW    NJPLNE=NUMBER OF WORDS                                                -
C                                                                           -
```

```
CD      JPLNES(I) NUMBER OF THE J-TH PLANE TO EDIT                      -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          KPLANES TO EDIT                                             -
C                                                                       -
CC     PRESENT IF NKIPLNE.GT.0                                          -
C                                                                       -
CL    (KPLNES(I),I=1,NKPLNE)                                            -
C                                                                       -
CW    NKPLNE=NUMBER OF WORDS                                            -
C                                                                       -
CD      KPLNES(I) NUMBER OF THE K-TH PLANE TO EDIT                      -
C                                                                       -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          CROSS SECTION POSITIONS TO EDIT                             -
C                                                                       -
CC     PRESENT IF NPOS.GT.0                                             -
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.0                                             -
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.0                                            -
C                                                                       -
CL    (EDMATS(I),I=1,MACRO)                                             -
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.0                                            -
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                                                                       -
C-----------------------------------------------------------------------
CR          RESPONSE FUNCTION NAMES                                     -
C                                                                       -
CC     PRESENT IF IDOSE.GT.0                                            -
C                                                                       -
CL    (RSFNAM(I),I=1,IDOSE)                                             -
C                                                                       -
CW    IDOSE*MULT=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.0                                            -
C                                                                       -
CL    (RSFE(I),I=1,NGROUP)                                              -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
CD      RSFE(I)   RESPONSE FOR GROUP I                                  -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          RESPONSE FUNCTION SPATIAL VECTOR IN X                       -
C                                                                       -
CC     PRESENT IF IDOSE.LT.0                                            -
C                                                                       -
CL    (RSFX(I),I=1,IT)                                                  -
C                                                                       -
CW    IT*MULT=NUMBER OF WORDS                                           -
C                                                                       -
CD      RSFX(I)   RESPONSE FUNCTION FOR FINE MESH I                     -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          RESPONSE FUNCTION SPATIAL VECTOR IN Y                       -
C                                                                       -
CC     PRESENT IF IDOSE.LT.0 .AND. IDIMEN.GT.1                          -
C                                                                       -
CL    (RSFY(J),J=1,JT)                                                  -
C                                                                       -
CW    JT*MULT=NUMBER OF WORDS                                           -
C                                                                       -
CD      RSFY(J)   RESPONSE FUNCTION FOR FINE MESH J                     -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          RESPONSE FUNCTION SPATIAL VECTOR IN Z                       -
C                                                                       -
CC     PRESENT IF IDOSE.LT.0 .AND. IDIMEN.GT.2                          -
C                                                                       -
CL    (RSFZ(K),K=1,KT)                                                  -
C                                                                       -
CW    KT*MULT=NUMBER OF WORDS                                           -
```

```
C
CD      RSFZ(K)    RESPONSE FUNCTION FOR FINE MESH K
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           CROSS SECTION SUMMING ARRAY
C
CC     PRESENT IF IXSUM.GT.0
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.0
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 VECTOR IN X
C
CC     PRESENT IF IDEN.GT.0
C
CL    (XDF(I),I=1,IT)
C
CW    IT*MULT=NUMBER OF WORDS
C
CD      XDF(I) DENSITY FACTOR FOR THE I-TH FINE MESH
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           FINE MESH DENSITY VECTOR IN Y
C
CC     PRESENT IF IDEN.GT.0 .AND. IDIMEN.GT.1
C
CL    (YDF(J),J=1,JT)
C
CW    JT*MULT=NUMBER OF WORDS
C
CD      YDF(J) DENSITY FACTOR FOR THE J-TH FINE MESH
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           FINE MESH DENSITY VECTOR IN Z
C
CC     PRESENT IF IDEN.GT.0 .AND. IDIMEN.GT.2
C
CL    (ZDF(K),K=1,KT)
C
CW    KT*MULT=NUMBER OF WORDS
```

```
C                                                                       -
CD      ZDF(K) DENSITY FACTOR FOR THE K-TH FINE MESH                    -
C                                                                       -
C-----------------------------------------------------------------------
CEOF                                                                    -
```

FISSRC

The FISSRC file is a binary, code-dependent file containing the energy-group total fission source at each spatial fine-mesh point, i, that is,

$$\sum_{g} (\nu\Sigma_f)_{g,i} \phi_{g,i} \ ,$$

The FISSRC file is automatically produced by the Solver Module whenever fissions are present .

```
C*****************************************************************-
C                        DATE 02/21/95                            -
C                                                                 -
CF          FISSRC                                                -
CE          CODE DEPENDENT FISSION SOURCE                         -
C                                                                 -
C*****************************************************************-

C-----------------------------------------------------------------
C                                                                 -
C                                                                 -
CN                THIS FILE PROVIDES THE FISSION SOURCE           -
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 - FISSRC - (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-------------------------------------------------------------------------
CR          SPECIFICATIONS      (1D RECORD)                              -
C                                                                        -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,ITER,EFFK,POWER,NBLOK                -
C                                                                        -
CW    9 =NUMBER OF WORDS                                                 -
C                                                                        -
CD    NDIM              NUMBER OF DIMENSIONS                             -
CD    NGROUP            NUMBER OF ENERGY GROUPS                          -
CD    NINTI             NUMBER OF FIRST DIMENSION FINE MESH INTERVALS    -
CD    NINTJ             NUMBER OF SECOND DIMENSION FINE MESH INTERVALS   -
CD    NINTK             NUMBER OF THIRD DIMENSION FINE MESH INTERVALS    -
CD    ITER              OUTER ITERATION NUMBER AT WHICH FISSION WAS       -
CD                      WRITTEN                                          -
CD    EFFK              EFFECTIVE MULTIPLICATION FACTOR                  -
CD    POWER             POWER IN WATTS TO WHICH FISSION IS NORMALIZED     -
CD    NBLOK             SET TO 1                                         -
C                                                                        -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          FISSION SOURCE      (2D RECORD)                              -
C                                                                        -
CL    ((FISS(I,J),I=1,NINTI),J=1,NINTJ)                                  -
C                                                                        -
CW    NINTI*NINTJ*MULT=NUMBER OF WORDS                                   -
C                                                                        -
C       DO 1 K=1, NINTK                                                  -
C     1 READ (N) *LIST AS ABOVE*                                         -
C                                                                        -
CD    FISS(I,J)           FISSION SOURCE WITHOUT VOLUME AT FINE MESH     -
CD                        POINT (I,J)IN PLANE K, I.E., NUSIGF*FLUX       -
C                                                                        -
C-------------------------------------------------------------------------
CEOF
```

GEODST

This GEODST file is an extended version of the standard GEODST file. It is a binary, code-dependent file containing the geometry description. This version is necessary for and is used only by the TWODANT/GQ module to describe the geometries based on generalized quadrilaterals used by that module. The other calculational modules use the standard GEODST file. The standard GEODST file is a subset of this extended version.

```
C*************************************************************************-
C   REVISED 11/30/76   EXTENDED 11/14/90  EXTENSION REVISED 05/20/91    -
C                                                                        -
CF          GEODST - IV                                                  -
C                                                                        -
CE          GEOMETRY DESCRIPTION                                         -
C                                                                        -
C*************************************************************************


C-------------------------------------------------------------------------
CS     FILE STRUCTURE                                                    -
CS                                                                       -
CS        RECORD TYPE                          PRESENT IF                -
CS        ========================== -        ==========                -
CS        FILE IDENTIFICATION                  ALWAYS                    -
CS        FILE SPECIFICATIONS                  ALWAYS                    -
CS        ONE DIMENSIONAL COARSE MESH          IGOM.GT.0 AND IGOM.LE.3   -
CS        TWO DIMENSIONAL COARSE MESH          IGOM.GE.6 AND IGOM.LE.11  -
CS        THREE DIMENSIONAL COARSE MESH        IGOM.GE.12 AND IGOM.LE.18 -
CS        GEOMETRY DATA                        IGOM.GT.0 OR NBS.GT.0     -
CS        REGION ASSIGNMENTS TO COARSE MESH    IGOM.GT.0 AND NRASS.EQ.0  -
CS        REGION ASSIGNMENTS TO FINE MESH      IGOM.GT.0 AND NRASS.EQ.1  -
CS                                                                       -
CS        GENERALIZED 2D SPECIFICATIONS        IGOM.EQ.106 OR IGOM.EQ.107 -
CS        SUBMESH JOINING DATA                 IGOM.EQ.106 OR IGOM.EQ.107 -
CS   ********(REPEAT FOR ALL SUB-MESHES)       IGOM.EQ.106 OR IGOM.EQ.107 -
CS   *    SUB-MESH NAME                                                  -
CS   *    SUB-MESH SPECIFICATIONS                                        -
CS   *    SUB-MESH VERTICES                                              -
CS   *    SUB-MESH INTERVAL CORNER COUNTS      ILR.EQ.0                  -
CS   *    SUB-MESH INTERVALS                   ILR.EQ.0                  -
CS   ********(END REPEAT)                                                -
CS   ********(REPEAT FOR ALL ZONINGS)          IGOM.EQ.106 OR IGOM.EQ.107 -
CS   *    ZONING NAME                          NRASS.EQ.2                -
CS   *    ZONING SPECIFICATIONS                NRASS.EQ.2                -
CS   *    REGION ASSIGNMENTS TO SUB-MESH       NRASS.EQ.2                -
CS   ********(END REPEAT)                                                -
CS        OBJECT NAMES                         IGOM.EQ.106 OR IGOM.EQ.107 -
CS        OBJECT SPECIFICATIONS                IGOM.EQ.106 OR IGOM.EQ.107 -
CS   ********(REPEAT FOR ALL COMPONENTS)       IGOM.EQ.106 OR IGOM.EQ.107 -
CS   *    COMPONENT NAME                                                 -
CS   *    COMPONENT SPECIFICATIONS                                       -
CS   *    COMPONENT DESCRIPTION                                          -
CS   ********(END REPEAT)                                                -
CS        GEOMETRY NAME                        IGOM.EQ.106 OR IGOM.EQ.107 -
CS        GEOMETRY SPECIFICATIONS              IGOM.EQ.106 OR IGOM.EQ.107 -
CS        GEOMETRY DESCRIPTION                 IGOM.EQ.106 OR IGOM.EQ.107 -
CS   ********(REPEAT FOR ALL BDRY SEGMENTS)    IGOM.EQ.106 OR IGOM.EQ.107 -
CS   *    BOUNDARY SEGMENT, CONDITION                                    -
CS   ********(END REPEAT)                                                -
C                                                                        -
```

```
C-------------------------------------------------------------------

C-------------------------------------------------------------------
CR          FILE IDENTIFICATION (0V RECORD)                        -
C                                                                  -
CL    HNAME,(HUSE(I),I=1,2),IVERS                                  -
C                                                                  -
CW    1+3*MULT                                                     -
C                                                                  -
CD    HNAME        HOLLERITH FILE NAME - GEODST - (A6)             -
CD    HUSE         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 SPECIFICATIONS (1D RECORD)                        -
C                                                                  -
CL    IGOM,NZONE,NREG,NZCL,NCINTI,NCINTJ,NCINTK,NINTI,NINTJ,NINTK,IMB1,-
CL    IMB2,JMB1,JMB2,KMB1,KMB2,NBS,NBCS,NIBCS,NZWBB,NTRIAG,NRASS,NTHPT,-
CL    (NGOP(I),I=1,4)                                              -
C                                                                  -
CW    27                                                           -
C                                                                  -
CD    IGOM         GEOMETRY  0- POINT (FUNDAMENTAL MODE)           -
CD                           1- SLAB                               -
CD                           2- CYLINDER                           -
CD                           3- SPHERE                             -
CD                           6- X-Y                                -
CD                           7- R-Z                                -
CD                           8- THETA-R                            -
CD                           9- UNIFORM TRIANGULAR                 -
CD                          10- HEXAGONAL (1 MESH POINT IN EACH    -
CD                                         HEXAGONAL ELEMENT)      -
CD                          11- R-THETA                            -
CD                          12- R-THETA-Z                          -
CD                          13- R-THETA-ALPHA                      -
CD                          14- X-Y-Z                              -
CD                          15- THETA-R-Z                          -
CD                          16- THETA-R-ALPHA                      -
CD                          17- UNIFORM TRIANGULAR-Z               -
CD                          18- HEXAGON-Z    (MESH POINTS AS IN 10 -
CD                                            ABOVE)               -
CD                         106- X-Y GENERALIZED                    -
CD                         107- R-Z GENERALIZED                    -
C                                                                  -
CD    NZONE        NUMBER OF ZONES (EACH HOMOGENEOUS IN NEUTRONICS -
CD                 PROBLEM - A ZONE CONTAINS ONE OR MORE REGIONS)  -
CD    NREG         NUMBER OF REGIONS                               -
CD    NZCL         NUMBER OF ZONE CLASSIFICATIONS (EDIT PURPOSES)  -
CD    NCINTI       NUMBER OF FIRST DIMENSION COARSE MESH INTERVALS -
CD    NCINTJ       NUMBER OF SECOND DIMENSION COARSE MESH          -
CD                     INTERVALS. NCINTJ.EQ.1 FOR ONE DIMENSIONAL  -
CD                     CASE.                                       -
CD    NCINTK       NUMBER OF THIRD DIMENSION COARSE MESH INTERVALS -
CD                     NCINTK.EQ.1 FOR ONE AND TWO DIMENSIONAL     -
CD                     CASES.                                      -
CD    NINTI        NUMBER OF FIRST DIMENSION FINE MESH INTERVALS   -
CD    NINTJ        NUMBER OF SECOND DIMENSION FINE MESH INTERVALS  -
CD                     NINTJ.EQ.1 FOR ONE DIMENSIONAL CASE.        -
```

```
CD    NINTK       NUMBER OF THIRD DIMENSION FINE MESH INTERVALS        -
CD                    NINTK.EQ.1 FOR ONE AND TWO DIMENSION CASES.     -
CD    IMB1        FIRST BOUNDARY ON FIRST DIMENSION                    -
CD                    0 - ZERO FLUX (DIFFUSION)                       -
CD                    1 - REFLECTED                                   -
CD                    2 - EXTRAPOLATED (DIFFUSION - DEL PHI/PHI       -
CD                        = -C/D WHERE C IS GIVEN AS BNDC BELOW       -
CD                        AND D IS THE GROUP DIFFUSION CONSTANT,      -
CD                        TRANSPORT - NO RETURN).                     -
CD                    3 - REPEATING (PERIODIC) WITH OPPOSITE FACE     -
CD                    4 - REPEATING (PERIODIC) WITH NEXT ADJACENT     -
CD                        FACE.                                       -
CD                    5 - INVERTED REPEATING ALONG THIS FACE.         -
CD                        (180 DEGREE ROTATION)                       -
CD                    6 - ISOTROPIC RETURN (TRANSPORT)                -
C                                                                     -
CC          NOTE FOR REPEATING CONDITIONS (3,4,5) - LET I1 DENOTE FIRST-
CC          BOUNDARY ON FIRST DIMENSION, I2 THE SECOND BOUNDARY ON THE -
CC          FIRST DIMENSION, J1 THE FIRST BOUNDARY ON THE SECOND      -
CC          DIMENSION, ETC. THEN THESE REPEATING BOUNDARY CONDITIONS   -
CC          ONLY APPLY TO BOUNDARIES I1,I2,J1, AND J2. GOING IN ORDER -
CC          OF I1,J1,I2,J2, THE FIRST BOUNDARY WHICH IS INVOLVED       -
CC          CARRIES THE DESIGNATOR DEFINING THE REPEATING CONDITION.   -
C                                                                     -
C                                                                     -
CD    IMB2        LAST BOUNDARY ON FIRST DIMENSION                     -
CD    JMB1        FIRST BOUNDARY ON SECOND DIMENSION                   -
CD    JMB2        LAST BOUNDARY ON SECOND DIMENSION                    -
CD    KMB1        FIRST BOUNDARY ON THIRD DIMENSION                    -
CD    KMB2        LAST BOUNDARY ON THIRD DIMENSION                     -
CD    NBS         NUMBER OF BUCKLING SPECIFICATIONS                    -
CD                    0 - NONE                                        -
CD                    1 - SINGLE VALUE APPLIES EVERYWHERE             -
CD                    .EQ.NZONE-  ZONE DEPENDENT                      -
CD                    M*NZONE - DATA IS GIVEN OVER ALL ZONES FOR      -
CD                        THE FIRST ENERGY GROUP, THEN FOR THE        -
CD                        NEXT GROUP, TO END OF LIST.  IF             -
CD                        M.LT.NGROUP THEN THE M-TH GROUP DATA        -
CD                        APPLIES TO ALL ADDITIONAL GROUPS.           -
CD                        (2.LE.M.LE.NGROUP)                          -
CD    NBCS        NUMBER OF CONSTANTS FOR EXTERNAL BOUNDARIES          -
CD                    0 - NONE                                        -
CD                    1 - SINGLE VALUE USED EVERYWHERE                -
CD                    6 - INDIVIDUAL VALUE GIVEN FOR EACH             -
CD                        EXTERNAL BOUNDARY. THE ORDERING OF THE      -
CD                        VALUES IS THE SAME AS THE ORDERING OF       -
CD                        THE BOUNDARY CONDITIONS.                    -
CD                    6*M - SIX VALUES GIVEN FOR FIRST ENERGY         -
CD                        GROUP (ORDERED AS DESCRIBED ABOVE),         -
CD                        THEN 6 FOR THE NEXT GROUP, TO END OF        -
CD                        LIST.  (2.LE.M.LE.NGROUP).                  -
CD                        IF M.LT.NGROUP THEN THE M-TH GROUP DATA     -
CD                        APPLIES TO ALL REMAINING GROUPS.            -
CD    NIBCS       NUMBER OF CONSTANTS FOR INTERNAL BOUNDARIES          -
CD                    0 - NONE                                        -
CD                    1 - SINGLE VALUE USED EVERYWHERE                -
CD                    .GT.1 - VALUES ARE GIVEN BY ENERGY GROUP        -
CD                      WITH NON-BLACK CONDITION INDICATED BY         -
CD                      ZERO ENTRY - LAST VALUE APPLIES TO            -
CD                      ADDITIONAL GROUPS                             -
CD    NZWBB       NUMBER OF ZONES WHICH ARE BLACK ABSORBERS            -
CD    NTRIAG      TRIANGULAR/HEXAGONAL GEOMETRY OPTION                 -
CD                    0 - REGION OF SOLUTION IS A RHOMBUS IN          -
CD                        WHICH THE 1ST AND 2ND DIMENSION AXES        -
CD                        INTERSECT AT AN ANGLE OF 120 DEGREES.       -
```

```
CD                    1 - REGION OF SOLUTION IS A RHOMBUS IN          -
CD                        WHICH THE 1ST AND 2ND DIMENSION AXES        -
CD                        INTERSECT AT AN ANGLE OF 60 DEGREES.        -
CD                    2 - REGION OF SOLUTION IS A RECTANGLE.  THE     -
CD                        BOUNDARIES I1 AND I2 BISECT MESH            -
CD                        TRIANGLES.  SEE NTHPT BELOW.                -
CD                        (IGOM=9,17 ONLY)                            -
CD                    3 - REGION OF SOLUTION IS AN EQUILATERAL,       -
CD                        60 DEGREE TRIANGLE.  (IGOM=9,17 ONLY)       -
CD                    4 - REGION OF SOLUTION IS A 30-60 DEGREE        -
CD                        RIGHT TRIANGLE IN WHICH THE 1ST AND 2ND     -
CD                        DIMENSION AXES INTERSECT AT THE 30          -
CD                        DEGREE ANGLE.  (IGOM=9,17 ONLY)             -
CD                    5 - REGION OF SOLUTION IS A RHOMBUS IN          -
CD                        WHICH THE 1ST AND 2ND DIMENSION AXES        -
CD                        INTERSECT AT AN ANGLE OF 30 DEGREES.        -
CD                        (IGOM=9,17 ONLY)                            -
CD    NRASS       REGION ASSIGNMENTS                                  -
CD                    0- TO COARSE MESH                               -
CD                    1- TO FINE MESH                                 -
CD                    2- TO SUBMESH                                   -
CD    NTHPT       ORIENTATION OF FIRST FINE MESH INTERVAL IN          -
CD                    TRIANGULAR GEOMETRIES.  NTRIAG=2 ONLY.          -
CD                    1- TRIANGLE(1,1) POINTS AWAY FROM FIRST         -
CD                       DIMENSION AXIS, I.E., NO INTERNAL MESH       -
CD                       LINE INTERSECTS THE ORIGIN.                  -
CD                    2- TRIANGLE(1,1) POINTS TOWARD THE FIRST        -
CD                       DIMENSION AXIS, I.E., AN INTERNAL MESH       -
CD                       LINE INTERSECTS THE ORIGIN.                  -
CD    NGOP        RESERVED                                            -
C                                                                     -
C-------------------------------------------------------------------------
```

```
C-------------------------------------------------------------------------
CR          ONE DIMENSIONAL COARSE MESH INTERVAL BOUNDARIES AND FINE   -
CR          MESH INTERVALS (2D RECORD)                                 -
C                                                                      -
CC          PRESENT IF IGOM.GT.0 AND IGOM.LE.3                         -
C                                                                      -
CL    (XMESH(I),I=1,NCBNDI),(IFINTS(I),I=1,NCINTI)                     -
C                                                                      -
CW    NCBNDI*MULT+NCINTI                                               -
C                                                                      -
CD    XMESH       COARSE MESH BOUNDARIES, FIRST DIMENSION              -
CD    IFINTS      NUMBER OF EQUALLY SPACED FINE MESH INTERVALS         -
CD                    PER COARSE MESH INTERVAL, FIRST DIMENSION.       -
CD    NCBNDI      NCINTI+1, NUMBER OF FIRST DIMENSION COARSE MESH      -
CD                    BOUNDARIES                                       -
C                                                                      -
CC        UNITS ARE CM FOR LINEAR DIMENSIONS AND RADIANS FOR ANGULAR   -
CC        DIMENSIONS                                                   -
C                                                                      -
C-------------------------------------------------------------------------
```

```
C-------------------------------------------------------------------------
CR          TWO DIMENSIONAL COARSE MESH INTERVAL BOUNDARIES AND FINE   -
CR          MESH INTERVALS (3D RECORD)                                 -
C                                                                      -
CC          PRESENT IF IGOM.GE.6 AND IGOM.LE.11                        -
C                                                                      -
CL    (XMESH(I),I=1,NCBNDI),(YMESH(J),J=1,NCBNDJ),                     -
CL   1(IFINTS(I),I=1,NCINTI),(JFINTS(J),J=1,NCINTJ)                    -
C                                                                      -
```

```
CW     (NCBNDI+NCBNDJ)*MULT+NCINTI+NCINTJ                                    -
C                                                                            -
CD     YMESH        COARSE MESH BOUNDARIES, SECOND DIMENSION                 -
CD     JFINTS       NUMBER OF EQUALLY SPACED FINE MESH INTERVALS             -
CD                      PER COARSE MESH INTERVAL, SECOND DIMENSION.          -
CD     NCBNDJ       NCINTJ+1, NUMBER OF SECOND DIMENSION COARSE              -
CD                      MESH BOUNDARIES                                      -
C                                                                            -
CC           FOR UNIFORM-TRIANGULAR-MESH GEOMETRY (IGOM = 9) THE             -
CC           LENGTH (L) OF THE SIDE OF A MESH TRIANGLE MUST BE GIVEN         -
CC           BY THE EXPRESSION                                               -
CC                 L = 2.*(XMESH(2)-XMESH(1))/IFINTS(1) .                    -
CC           FOR UNIFORM-HEXAGONAL-MESH GEOMETRY (IGOM = 10) THE             -
CC           FLAT-TO-FLAT DISTANCE (FTF) ACROSS A MESH HEXAGON MUST          -
CC           BE GIVEN BY THE EXPRESSION                                      -
CC                 FTF = (XMESH(2)-XMESH(1))/IFINTS(1)                       -
C                                                                            -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR           THREE DIMENSIONAL COARSE MESH INTERVAL BOUNDARIES AND FINE      -
CR           MESH INTERVALS (4D RECORD)                                      -
C                                                                            -
CC           PRESENT IF IGOM.GE.12 .AND. IGOM.LT.100                         -
C                                                                            -
CL     (XMESH(I),I=1,NCBNDI),(YMESH(J),J=1,NCBNDJ),                          -
CL    1(ZMESH(K),K=1,NCBNDK),(IFINTS(I),I=1,NCINTI),                         -
CL    2(JFINTS(J),J=1,NCINTJ),(KFINTS(K),K=1,NCINTK)                         -
C                                                                            -
CW     (NCBNDI+NCBNDJ+NCBNDK)*MULT+NCINTI+NCINTJ+NCINTK                      -
C                                                                            -
CD     ZMESH        COARSE MESH BOUNDARIES, THIRD DIMENSION                  -
CD     KFINTS       NUMBER OF EQUALLY SPACED FINE MESH INTERVALS             -
CD                      PER COARSE MESH INTERVAL, THIRD DIMENSION.           -
CD     NCBNDK       NCINTK+1, NUMBER OF THIRD DIMENSION COARSE MESH          -
CD                      BOUNDARIES                                           -
C                                                                            -
CC           FOR UNIFORM-TRIANGULAR-MESH GEOMETRY (IGOM = 17) THE            -
CC           LENGTH (L) OF THE SIDE OF A MESH TRIANGLE MUST BE GIVEN         -
CC           BY THE EXPRESSION                                               -
CC                 L = 2.*(XMESH(2)-XMESH(1))/IFINTS(1) .                    -
CC           FOR UNIFORM-HEXAGONAL-MESH GEOMETRY (IGOM = 18) THE             -
CC           FLAT-TO-FLAT DISTANCE (FTF) ACROSS A MESH HEXAGON MUST          -
CC           BE GIVEN BY THE EXPRESSION                                      -
CC                 FTF = (XMESH(2)-XMESH(1))/IFINTS(1)                       -
C                                                                            -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR           GEOMETRY DATA (5D RECORD)                                       -
C                                                                            -
CC           PRESENT IF IGOM.GT.0 OR NBS.GT.0                                -
C                                                                            -
CL     (VOLR(N),N=1,NREG),(BSQ(N),N=1,NBS),(BNDC(N),N=1,NBCS),               -
CL     (BNCI(N),N=1,NIBCS),((NZHBB(N),N=1,NZWBB),(NZC(N),N=1,NZONE),         -
CL     (NZNR(N),N=1,NREG)                                                    -
C                                                                            -
CW     2*NREG+NBS+NBCS+NIBCS+NZWBB+NZONE                                     -
C                                                                            -
CD     VOLR         REGION VOLUMES (CC)                                     -
CD     BSQ          BUCKLING (B**2) VALUES (CM**-2)                         -
CD     BNDC         BOUNDARY CONSTANTS (DEL PHI/PHI =-C/D)                  -
CD     BNCI         INTERNAL BLACK BOUNDARY CONSTANTS                       -
CD     NZHBB        ZONE NUMBERS WITH BLACK ABSORBER CONDITIONS             -
CD     NZC          ZONE CLASSIFICATIONS                                    -
```

```
CD    NZNR        ZONE NUMBER ASSIGNED TO EACH REGION                    -
C                                                                        -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          REGION ASSIGNMENTS TO COARSE MESH INTERVALS (6D RECORD)       -
C                                                                        -
CC          PRESENT IF IGOM.GT.0 AND NRASS.EQ.0                          -
C                                                                        -
CL    ((MR(I,J),I=1,NCINTI),J=1,NCINTJ)----NOTE STRUCTURE BELOW----      -
C                                                                        -
CW    NCINTI*NCINTJ                                                      -
C                                                                        -
CS    DO 1 K=1,NCINTK                                                    -
CS  1 READ(N) *LIST AS ABOVE*                                            -
C                                                                        -
CD    MR          REGION NUMBERS ASSIGNED TO COARSE MESH                 -
CD                    INTERVALS                                          -
C                                                                        -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          REGION ASSIGNMENTS TO FINE MESH INTERVALS (7D RECORD)        -
C                                                                        -
CC          PRESENT IF IGOM.GT.0 AND NRASS.EQ.1                          -
C                                                                        -
CL    ((MR(I,J),I=1,NINTI),J=1,NINTJ)----NOTE STRUCTURE BELOW----        -
C                                                                        -
CW    NINTI*NINTJ                                                        -
C                                                                        -
CS    DO 1 K=1,NINTK                                                     -
CS  1 READ(N) *LIST AS ABOVE*                                            -
C                                                                        -
CD    MR          REGION NUMBERS ASSIGNED TO FINE MESH INTERVALS         -
C                                                                        -
C-------------------------------------------------------------------------


CC    THE FULL PROBLEM AREA MAY BE DECOMPOSED INTO A SET OF SEPARATE     -
CC  SMALLER AREAS, EACH WITH A MESH OF ITS OWN, CALLED A SUBMESH.        -
CC  EACH SUB-MESH IS THEN DESCRIBED SEPARATELY. SOME VERTICES ON THE     -
CC  EXTERIOR OF A SUBMESH MAY ALSO LIE IN A CONTIGUOUS SUBMESH.          -
CC                                                                       -
CC    NOTICE THERE IS NO IMPLIED CONSTRAINT THAT EACH EXTERIOR SUBMESH   -
CC  VERTEX MATCH A VERTEX ON AN ADJACENT SUBMESH. THUS, ONE SUBMESH MAY- 
CC  BE FINER THAN ITS ADJOINING ONE.                                     -
CC                                                                       -

C-------------------------------------------------------------------------
CR          GENERALIZED 2D SPECIFICATIONS (8D RECORD)                    -
C                                                                        -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                        -
C                                                                        -
CL    NSMESH,NZONIG,NOBJS,NCOMPS,NEB,                                    -
CL    MAXVTX,MAXICN,MAXINT,NUMVTX,NUMICN,NUMINT,NUMCOM,NCOMGE,NAMLEN     -
C                                                                        -
CW    14                                                                 -
C                                                                        -
CD    NSMESH      NUMBER OF SUBMESHES                                    -
CD    NZONIG      NUMBER OF ZONINGS                                      -
CD    NOBJS       NUMBER OF OBJECTS                                      -
CD    NCOMPS      NUMBER OF COMPONENTS IN GEOMETRY                       -
CD    NEB         NUMBER OF EXTERNAL BOUNDARY SEGMENTS                   -
```

```
CD    MAXVTX      MAXIMUM NUMBER OF VERTEXES IN A SUBMESH                -
CD    MAXICN      MAXIMUM NUMBER OF INTERVAL CORNERS IN A SUBMESH        -
CD    MAXINT      MAXIMUM NUMBER OF INTERVALS IN A SUBMESH               -
CD    NUMVTX      TOTAL NUMBER OF VERTEXES IN ALL SUBMESHES              -
CD    NUMICN      TOTAL NUMBER OF INTERVAL CORNERS IN ALL SUBMESHES      -
CD    NUMINT      TOTAL NUMBER OF INTERVALS IN ALL SUBMESHES             -
CD    NUMCOM      TOTAL NUMBER OF SUBCOMPONENTS IN ALL COMPONENTS        -
CD                AND THE GEOMETRY                                       -
CD    NCOMGE      TOTAL NUMBER OF SUBCOMPONENTS IN GEOMETRY              -
CD    NAMLEN      NUMBER OF A6 WORDS IN A SUBMESH, OBJECT, OR            -
CD                COMPONENT NAME                                         -
C                                                                        -
C------------------------------------------------------------------------

C------------------------------------------------------------------------
CR          SUBMESH JOINING DATA (9D RECORD)                             -
C                                                                        -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                        -
C                                                                        -
CL    XYEPS                                                              -
C                                                                        -
CW    1*MULT = NUMBER OF WORDS                                           -
C                                                                        -
CD    XYEPS       EPSILON FOR POSITION IDENTITY                          -
C                                                                        -
CC NOTE: WHEN JOINING TWO SUBMESHES TO FORM A LARGER MESH, SOME          -
CC       INTERFACE VERTEXES MAY BE DESCRIBED TWICE, ONCE IN EACH         -
CC       SUBMESH. IF THEY LIE WITHIN XYEPS OF EACH OTHER, THEY MAY       -
CC       BE CONSIDERED TO BE THE SAME VERTEX.                            -
C------------------------------------------------------------------------

C------------------------------------------------------------------------
CR          SUB-MESH NAME (10D RECORD)                                   -
C                                                                        -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                        -
C                                                                        -
CL    SNAME                                                              -
C                                                                        -
CW    1*MULT*NAMLEN = NUMBER OF WORDS                                    -
C                                                                        -
CD    SNAME       HOLLERITH SUB-MESH NAME (A6)                           -
C                                                                        -
C------------------------------------------------------------------------

C------------------------------------------------------------------------
CR          SUB-MESH SPECIFICATIONS (11D RECORD)                         -
C                                                                        -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                        -
C                                                                        -
CL    NVTX,ILR,IM,JM,NEDGES,NINT                                         -
C                                                                        -
CW    6 = NUMBER OF WORDS                                                -
C                                                                        -
CD    NVTX        NUMBER OF VERTICES IN THIS SUB-MESH                    -
CD    ILR         FLAG FOR LOGICALLY RECTANGULAR SUB-MESH DOMAIN         -
CD                    0 - NOT A LOGICALLY RECTANGULAR DOMAIN             -
CD                    1 - LOGICALLY RECTANGULAR DOMAIN                   -
CD    IM          NUMBER OF VERTICES IN THE X DIRECTION IF THE           -
CD                SUB-MESH HAS A LOGICALLY RECTANGULAR DOMAIN            -
CD    JM          NUMBER OF VERTICES IN THE Y DIRECTION IF THE           -
CD                SUB-MESH HAS A LOGICALLY RECTANGULAR DOMAIN            -
CD    NICNT       TOTAL NUMBER OF INTERVAL CORNERS IN THE SUB-MESH       -
CD    NINT        NUMBER OF MESH INTERVALS IN THE SUB-MESH               -
C                                                                        -
C------------------------------------------------------------------------
```

```
C-------------------------------------------------------------------------
CR          SUB-MESH VERTEXES (12D RECORD)                              -
C                                                                       -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                       -
C                                                                       -
CL    (XVERT(I),I=1,NVTX),(YVERT(I),I=1,NVTX)                           -
C                                                                       -
CW    2*NVTX*MULT = NUMBER OF WORDS                                     -
C                                                                       -
CD    XMESH       X COORDINATES OF VERTICES                             -
CD    YMESH       Y COORDINATES OF VERTICES                             -
C                                                                       -
C-------------------------------------------------------------------------


C      FOR A LOGICALLY RECTANGULAR DOMAIN, THE EDGES AND                -
C      INTERVALS ARE AUTOMATICALLY DEFINED. THE FIRST IM+1              -
C      VERTEXES ABOVE FORM, IN ORDER,THE FIRST LINE IN THE FIRST        -
C      DIMENSION, THE SECOND IM+1 POINTS FORM THE SECOND LINE.          -
C      A LINE CONTAINS IM EDGES, EACH FORMED BY CONNECTING              -
C      SUCCESSIVE VERTEXES AND EACH VERTEX IN THE FIRST LINE IS         -
C      CONNECTED BY AN EDGE TO THE CORRESPONDING VERTEX IN THE          -
C      SECOND LINE. THIS SERIES OF EDGES ENCLOSES THE FIRST IM          -
C      INTERVALS. CONTINUING THIS PROCESS THRU ALL THE LINES            -
C      GENERATES THE FULL LOGICALLY RECTANGULAR MESH STRUCTURE.         -


C-------------------------------------------------------------------------
CR          SUB-MESH INTERVAL CORNER COUNTS (13D RECORD)                -
C                                                                       -
CC          PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND ILR.EQ.0        -
C                                                                       -
CL    (NICN(I),I=1,NINT)                                                -
C                                                                       -
CW    NINT = NUMBER OF WORDS
C-------------------------------------------------------------------------
C                                                                       -
CD    NICN(I)     NUMBER OF CORNERS IN INTERVAL I                       -
C                                                                       -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          SUB-MESH INTERVALS (14D RECORD)                             -
C                                                                       -
CC          PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND ILR.EQ.0        -
C                                                                       -
CL    (ICN(I),I=1,NICNT)                                                -
C                                                                       -
CW    NICNT = NUMBER OF WORDS                                           -
C                                                                       -
CD    ICN(I)      INDEXES OF VERTEXES SURROUNDING EACH INTERVAL.        -
CD                THE FIRST NICN(1) OF THESE DESCRIBE THE               -
CD                VERTEXES SURROUNDING INTERVAL NUMBER ONE. THE         -
CD                NEXT NICN(2) DESCRIBE THE VERTEXES AROUND             -
CD                INTERVAL NUMBER TWO, AND SO ON.                       -
C                                                                       -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          ZONING NAME (15D RECORD)                                    -
```

```
C                                                                       -
CC          PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND NRASS.EQ.2      -
C                                                                       -
CL    ZNAME                                                             -
C                                                                       -
CW    1*MULT*NAMLEN = NUMBER OF WORDS                                   -
C                                                                       -
CD    ZNAME       HOLLERITH ZONING NAME (A6)                            -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          ZONING SPECIFICATIONS (16D RECORD)                          -
C                                                                       -
CC          PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND NRASS.EQ.2      -
C                                                                       -
CL    LENZON                                                            -
C                                                                       -
CW    1 = NUMBER OF WORDS                                               -
C                                                                       -
CD    LENZON      NUMBER OF INTERVALS IN THIS ZONING                    -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          REGION ASSIGNMENTS BY SUBMESH INTERVAL (17D RECORD)         -
C                                                                       -
CC          PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND NRASS.EQ.2      -
C                                                                       -
CL    (MR(I),I=1,LENZON)                                                -
C                                                                       -
CW    LENZON = NUMBER OF WORDS                                          -
C                                                                       -
CD    MR          REGION NUMBERS ASSIGNED TO MESH INTERVALS             -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          OBJECT NAMES (18D RECORD)                                   -
C                                                                       -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                       -
C                                                                       -
CL    (OBNAME(I),I=1,NOBJS)                                             -
C                                                                       -
CW    NOBJS*MULT*NAMLEN = NUMBER OF WORDS                               -
C                                                                       -
CD    OBNAME      HOLLERITH OBJECT NAME (A6)                            -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          OBJECT SPECIFICATIONS (19D RECORD)                          -
C                                                                       -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                       -
C                                                                       -
CL    (KSMESH(I),I=1,NOBJS),(KZONIG(I),I=1,NOBJS)                       -
C                                                                       -
CW    2*NOBJS = NUMBER OF WORDS                                         -
C                                                                       -
CD    KSMESH      INDEX OF SUB-MESH FOR THIS OBJECT                     -
CD    KZONIG      INDEX OF ZONING FOR THIS OBJECT                       -
```

```
C                                                                              -
C------------------------------------------------------------------------------

C------------------------------------------------------------------------------
CR          COMPONENT NAME (20D RECORD)                                        -
C                                                                              -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                              -
C                                                                              -
CL    CMNAME                                                                   -
C                                                                              -
CW    MULT*NAMLEN = NUMBER OF WORDS                                            -
C                                                                              -
CD    CMNAME      HOLLERITH COMPONENT NAME (A6)                                -
C                                                                              -
C------------------------------------------------------------------------------

C------------------------------------------------------------------------------
CR          COMPONENT SPECIFICATIONS (21D RECORD)                              -
C                                                                              -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                              -
C                                                                              -
CL    NSCOMS                                                                   -
C                                                                              -
CW    1 = NUMBER OF WORDS                                                      -
C                                                                              -
CD    NSCOMS      NUMBER OF SUBCOMPONENTS IN COMPONENT                         -
C                                                                              -
C------------------------------------------------------------------------------

C------------------------------------------------------------------------------
CR          COMPONENT DESCRIPTION (22D RECORD)                                 -
C                                                                              -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                              -
C                                                                              -
CL    (X0(I),I=1,NSCOMS),(Y0(I),I=1,NSCOMS),(THETA(I),I=1,NSCOMS),             -
CL    (XTRANS(I),I=1,NSCOMS),(YTRANS(I),I=1,NSCOMS),                           -
CL    (IOBJ(I),I=1,NSCOMS)                                                     -
C                                                                              -
CW    5*NSCOMS*MULT+NSCOMS = NUMBER OF WORDS                                   -
C                                                                              -
CD    X0          X COORDINATE OF ROTATION POINT                               -
CD    Y0          Y COORDINATE OF ROTATION POINT                               -
CD    THETA       ROTATION ANGLE, CLOCKWISE DEGREES                            -
CD    XTRANS      X COORDINATE TRANSLATION DISTANCE                            -
CD    YTRANS      Y COORDINATE TRANSLATION DISTANCE                            -
CD    IOBJ        INDEX OF AN OBJECT OR COMPONENT                              -
CD                  POSITIVE FOR OBJECT                                        -
CD                  NEGATIVE FOR COMPONENT                                     -
C                                                                              -
C------------------------------------------------------------------------------

C------------------------------------------------------------------------------
CR          GEOMETRY NAME (23D RECORD)                                         -
C                                                                              -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                              -
C                                                                              -
CL    CMNAME                                                                   -
C                                                                              -
CW    MULT*NAMLEN = NUMBER OF WORDS                                            -
C                                                                              -
CD    GENAME      HOLLERITH GEOMETRY NAME (A6)                                 -
C                                                                              -
```

```
C-------------------------------------------------------------------------
C-------------------------------------------------------------------------
CR          GEOMETRY SPECIFICATIONS (24D RECORD)                        -
C                                                                       -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                       -
C                                                                       -
CL    NSCOMS                                                            -
C                                                                       -
CW    1 = NUMBER OF WORDS                                               -
C                                                                       -
CD    NSCOMS       NUMBER OF SUBCOMPONENTS IN GEOMETRY                  -
C                                                                       -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          GEOMETRY DESCRIPTION (25D RECORD)                           -
C                                                                       -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107                       -
C                                                                       -
CL    (X0(I),I=1,NSCOMS),(Y0(I),I=1,NSCOMS),(THETA(I),I=1,NSCOMS),      -
CL    (XTRANS(I),I=1,NSCOMS),(YTRANS(I),I=1,NSCOMS),                    -
CL    (IOBJ(I),I=1,NSCOMS)                                              -
C                                                                       -
CW    5*NSCOMS*MULT+NSCOMS = NUMBER OF WORDS                            -
C                                                                       -
CD    X0           X COORDINATE OF ROTATION POINT                       -
CD    Y0           Y COORDINATE OF ROTATION POINT                       -
CD    THETA        ROTATION ANGLE, CLOCKWISE DEGREES                    -
CD    XTRANS       X COORDINATE TRANSLATION DISTANCE                    -
CD    YTRANS       Y COORDINATE TRANSLATION DISTANCE                    -
CD    IOBJ         INDEX OF AN OBJECT OR COMPONENT                      -
CD                   POSITIVE FOR OBJECT                                -
CD                   NEGATIVE FOR COMPONENT                             -
C                                                                       -
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
CR          BOUNDARY SEGMENT, CONDITION (26D RECORD)                    -
C                                                                       -
CC          PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107 AND NEB.GT.0          -
C                                                                       -
CL    X1,Y1,X2,Y2,BTHETA,IB,IR                                          -
C                                                                       -
CW    5*MULT+2 = NUMBER OF WORDS                                        -
C                                                                       -
CD    X1           X COORDINATE OF STARTING POINT                       -
CD    Y1           Y COORDINATE OF STARTING POINT                       -
CD    X2           X COORDINATE OF ENDING POINT                         -
CD    Y2           Y COORDINATE OF ENDING POINT                         -
CD    BTHETA       COUNTERCLOCKWISE ANGLE BETWEEN THIS SEGMENT AND ITS  -
CD                 RECIPIENT SEGMENT (IF ANY). UNITS ARE DEGREES        -
CD    IB           BOUNDARY CONDITION                                   -
CD                 SAME OPTIONS AS IMB1 IN 1D RECORD, PLUS:             -
CD                   7 - EXITING ANGULAR FLUXES REENTER ELSEWHERE       -
CD                   8 - EXITING ANGULAR FLUXES ARE MIRROR              -
C                        IMAGED ABOUT THE PERPENDICULAR TO THE          -
C                        SEGMENT AND THEN REENTERED ELSEWHERE           -
CD    IR           INDEX OF THE BOUNDARY SEGMENT THAT IS THE            -
CD                 RECIPIENT OF FLUXES EXITING THIS SEGMENT             -
```

```
CD                  (NEEDED ONLY FOR IB=7 OR 8)                          -
C                                                                        -
C------------------------------------------------------------------------
```

GEOSING

This file communicates the geometry information from the Solver module of TWODANT/GQ to the EDIT module. It is of the form of an extended GEODST file, but consists of the single submesh and zoning that the Solver module assembled from the original, possibly multi-submesh, extended, GEODST file.

LNK3DNT

The LNK3DNT file is a binary, code dependent file to enable the mixing of macroscopic cross sections on the fine mesh by a volume fraction method. This is used in TWODANT and THREEDANT only in either X-Y or X-Y-Z symmetries respectively to model more complicated geometrical bodies in those symmetries.

```
C*****************************************************************-
C                        DATE 02/15/95                             -
C                                                                  -
CF          LNK3DNT                                                -
CE          FINE MESH MIXING FILE                                  -
C                                                                  -
C*****************************************************************-

C-----------------------------------------------------------------
C                                                                 -
C                                                                 -
CN    THIS FILE PROVIDES THE FINE MESH MIXING PRESCRITIONS        -
C                                                                 -
C                                                                 -
C-----------------------------------------------------------------

C-----------------------------------------------------------------
C                                                                 -
C                                                                 -
CD                       NOTE THAT DOUBLE PRECISION VOLUME FRACTIONS -
CD                       ARE GIVEN WHEN MULT=2                    -
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 - LNK3DNT -- (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-----------------------------------------------------------------
CR          SPECIFICATIONS      (1D RECORD)                       -
C                                                                 -
CL    NMXSP,NINTI,NINTJ,NINTK                                     -
C                                                                 -
CW    4 =NUMBER OF WORDS                                          -
C                                                                 -
CD    NMXSP              NUMBER OF MIXING INSTRUCTIONS            -
CD    NINTI              NUMBER OF FIRST DIMENSION FINE MESH INTERVALS -
CD    NINTJ              NUMBER OF SECOND DIMENSION FINE MESH INTERVALS -
```

```
CD    NINTK              NUMBER OF THIRD DIMENSION FINE MESH INTERVALS  -
C                                                                       -
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
CR          MIXING ARRAYS      (2D RECORDS)                             -
C                                                                       -
CL        (IDC(I),I=1,NMXSP)                                            -
C                                                                       -
CW    NMXSP=NUMBER OF WORDS                                             -
C                                                                       -
CL        (DEN(I),I=1,NMXSP)                                            -
C                                                                       -
CW    NMXSP*MULT=NUMBER OF WORDS                                        -
C                                                                       -
CL        (IPT(I),I=1,NMXSP)                                            -
C                                                                       -
CW    NMXSP=NUMBER OF WORDS                                             -
C                                                                       -
CD    ICD(I)          MACROSCOPIC MATERIAL NUMBER AT I                  -
CD    DEN(I)          VOLUME FRACTION AT I                              -
CD    IPT(I)          FINE MESH CELL NUMBER AT I                        -
C                                                                       -
C-----------------------------------------------------------------------
CEOF
```

MACRXS

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 05/12/83                          -
C                                                                 -
CF          MACRXS                                                -
CE          CODE DEPENDENT MACROSCOPIC MULTIGROUP CROSS SECTION FILE -
CE          FOR USE IN ONEDANT SOLVER MODULE                      -
C                                                                 -
C                                                                 -
C*****************************************************************-
C                                                                 -
CN                THIS FILE PROVIDES A BASIC BROAD GROUP          -
CN                LIBRARY, ORDERED BY GROUP                       -
C                                                                 -
C                 ORDER OF GROUPS IS ACCORDING TO DECREASING 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.0    -
CS   *         SCATTERING MATRIX                    NORD.NE.0    -
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 - MACRXS - (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-----------------------------------------------------------------
CR          FILE CONTROL                                          -
C                                                                 -
CL    NGROUP,NMAT,NORD,NED,IDPF,LNG,MAXUP,MAXDN,NPRIN,I2LP1       -
C                                                                 -
CW    10=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 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               -
CD    NPRIN          NUMBER OF PRINCIPAL CROSS SECTIONS                 -
CD    I2LP1          0/1 = NO/YES  2L+1 TERM WAS INCLUDED IN LIBRARY    -
C                                                                       -
C-----------------------------------------------------------------------
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)*MULT=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 BOUND OF GROUP N (EV)               -
CD    EMIN           MINIMUM ENERGY BOUND OF SET (EV)                   -
CD    NEDT           NED+NPRIN                                          -
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                                                                       -
CN                   ALSO PRESENT WHEN NPRIN=5 IS:                      -
C                                                                       -
CN                         HED(5) = 5HTRANS                           -
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*MULT=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                                                                       -
CN                   ALSO PRESENT WHEN NPRIN=5 IS:                      -
C                                                                       -
CN                        J=5  TRANSPORT CROSS SECTION                  -
C                                                                       -
C-----------------------------------------------------------------------
```

```
C                                                                       -
C-----------------------------------------------------------------------
CR          SCATTERING CONTROL BLOCK FOR GROUP N                        -
C                                                                       -
CC        PRESENT IF NORD.GT.0                                          -
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.0                                          -
C                                                                       -
CL    (SCAT(I),I=1,NTAB)                                                -
C                                                                       -
CW    NTAB*MULT=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                                                                    -
```

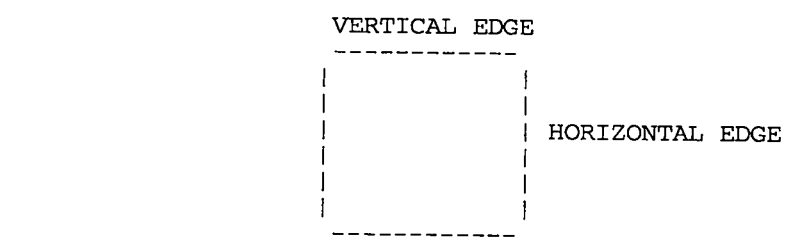
RAFLXM for TWODANT

The RAFLXM file is a binary, code-dependent file containing the angular fluxes at each fine-mesh boundary. It differs from the standard angular flux file RAFLUX only in that the fluxes are in different angular order. In RAFLXM, the fluxes are in calculational order according to the sweeping; i.e., angles for each level of a quadrant.

```
C*****************************************************************-
C                          DATE 12/13/85                          -
C                                                                 -
CF          RAFLXM                                                -
CE          CODE DEPENDENT COMPUTATIONAL-ORDERED ANGULAR FLUX AT CELL -
CE          EDGES FOR TWODANT CODE                                -
C                                                                 -
C*****************************************************************-


C-----------------------------------------------------------------
C                                                                 -
C                                                                 -
CN               THIS FILE PROVIDES THE EDGE ANGULAR FLUX AS ORDERED -
CN               BY THE CODE                                      -
C                                                                 -
C                                                                 -
C-----------------------------------------------------------------


C-----------------------------------------------------------------
C                                                                 -
CD                     ORDER OF GROUPS IS ACCORDING TO DECREASING  -
CD                     ENERGY. NOTE THAT DOUBLE PRECISION FLUXES ARE -
CD                     GIVEN WHEN MULT=2                          -
C                                                                 -
C-----------------------------------------------------------------


C-----------------------------------------------------------------
CS          FILE STRUCTURE                                        -
CS                                                                -
CS             RECORD TYPE                          PRESENT IF    -
CS             ============================         ==========    -
CS             FILE IDENTIFICATION                  ALWAYS        -
CS             FILE CONTROL                         ALWAYS        -
CS                                                                -
CS   ************(REPEAT FOR ALL GROUPS)                         -
CS   *              (GROUP 1 IS FIRST)                           -
CS   *   *********(REPEAT FOR ALL DIRECTIONS)                    -
CS   *   *     COMPUTATIONAL COSINES                ALWAYS        -
CS   *   *     HORIZONTAL-EDGE ANGULAR FLUX         ALWAYS        -
CS   *   *     VERTICAL-EDGE ANGULAR FLUX           ALWAYS        -
CS   *   *********                                               -
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 - RAFLXM - (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-----------------------------------------------------------------------
CR          SPECIFICATIONS      (1D RECORD)                             -
C                                                                       -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,EFFK,POWER                          -
C                                                                       -
CW    8 =NUMBER OF WORDS                                                -
C                                                                       -
CD    NDIM               NUMBER OF DIMENSIONS                           -
CD    NGROUP             NUMBER OF ENERGY GROUPS                        -
CD    NINTI              NUMBER OF FIRST DIMENSION FINE MESH INTERVALS  -
CD    NINTJ              NUMBER OF SECOND DIMENSION FINE MESH INTERVALS -
CD    NINTK              NUMBER OF THIRD DIMENSION FINE MESH INTERVALS. -
CD                       NINTK.EQ.1 IF NDIM.LE.2                        -
CD    EFFK               EFFECTIVE MULTIPLICATION FACTOR                -
CD    POWER              POWER IN WATTS TO WHICH FLUX IS NORMALIZED      -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          COMPUTATIONAL COSINES                                       -
C                                                                       -
CL    XMU,XETA,WEIGHT,XM                                                -
C                                                                       -
CW    4*MULT                                                            -
C                                                                       -
CD    XMU            MU COSINE                                          -
CD    XETA           ETA COSINE                                         -
CD    WEIGHT         WEIGHT                                             -
CD    XM             COMPRESSED COSINE TABLE INDEX                      -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          HORIZONTAL EDGE ANGULAR FLUX                                -
C                                                                       -
CL    ((HEDGE(I,J),I=NBDRYI),J=1,NINTJ)                                 -
C                                                                       -
CW    NBDRYI*NINTJ*MULT=NUMBER OF WORDS                                 -
C                                                                       -
CD    HEDGE(I,J)     HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX              -
C                                                                       -
CD    NBDRYI         NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH       -
CD                   BOUNDARIES)                                        -
C                                                                       -
C-----------------------------------------------------------------------

C-----------------------------------------------------------------------
CR          VERTICAL EDGE ANGULAR FLUX                                  -
C                                                                       -
```

```
CL     ((VEDGE(I,J),I=1,NINTI),J=1,NBDRYJ)                              -
C                                                                       -
CW     NINTI*NBDRYJ*MULT=NUMBER OF WORDS                                -
C                                                                       -
CD     VEDGE(I,J)      VERTICAL-EDGE-BOUNDARY ANGULAR FLUX              -
C                                                                       -
CD     NBDRYJ          NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH    -
CD                     BOUNDARIES)                                      -
C                                                                       -
C-----------------------------------------------------------------------
```




```
CEOF
```

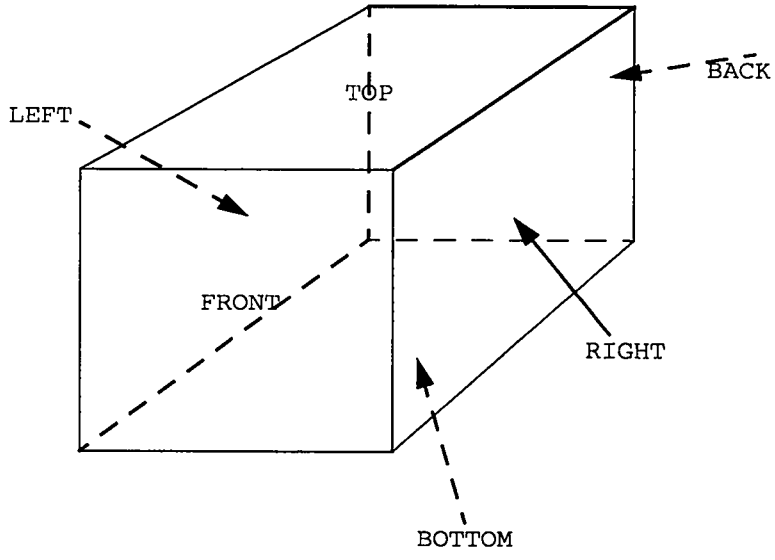
RAFLXM for THREEDANT

The RAFLXM file is a binary, code-dependent file containing the angular fluxes for the flux at each fine-mesh boundary. It differs from the standard angular flux file RAFLUX only in that the fluxes are in different order. In RAFLXM, the fluxes are in calculational order.

```
C*********************************************************************-
C                          DATE 03/03/95                              -
C                                                                     -
CF          RAFLXM                                                    -
CE          CODE DEPENDENT COMPUTATIONAL-ORDERED ANGULAR FLUX AT CELL  -
CE          EDGES FOR THREEDANT CODE                                  -
C                                                                     -
C*********************************************************************-


C---------------------------------------------------------------------
C                                                                     -
C                                                                     -
CN                THIS FILE PROVIDES THE EDGE ANGULAR FLUX AS ORDERED -
CN                BY THE CODE                                       -
C                                                                     -
C                                                                     -
C---------------------------------------------------------------------


C---------------------------------------------------------------------
C                                                                     -
CD                       ORDER OF GROUPS IS ACCORDING TO DECREASING   -
CD                       ENERGY. NOTE THAT DOUBLE PRECISION FLUXES ARE -
CD                       GIVEN WHEN MULT=2                            -
C                                                                     -
C---------------------------------------------------------------------


C---------------------------------------------------------------------
CS          FILE STRUCTURE                                            -
CS                                                                    -
CS             RECORD TYPE                          PRESENT IF        -
CS             ============================         ==========        -
CS             FILE IDENTIFICATION                  ALWAYS            -
CS             FILE CONTROL                         ALWAYS            -
CS                                                                    -
CS   *************(REPEAT FOR ALL GROUPS)                           -
CS   *               (GROUP 1 IS FIRST)                             -
CS   *     ********(REPEAT FOR ALL FRONT-GOING DIRECTIONS)           -
CS   *     *   COMPUTATIONAL COSINES                ALWAYS           -
CS   *     *   CELL BACK-EDGE ANGULAR FLUX FOR BACK PLANE            -
CS   *     *******                                                   -
CS   *                                                               -
CS   *     *******(REPEAT FOR ALL Z-INTERVALS)                       -
CS   *               (INTERVAL NINTK IS FIRST)                       -
CS   *     *  ******(REPEAT FOR ALL FRONT-GOING DIRECTIONS)          -
CS   *     *  * COMPUTATIONAL COSINES              ALWAYS           -
CS   *     *  * HORIZONTAL-EDGE ANGULAR FLUX       ALWAYS           -
CS   *     *  * VERTICAL-EDGE ANGULAR FLUX         ALWAYS           -
CS   *     *  * CELL FRONT-EDGE ANGULAR FLUX       ALWAYS           -
CS   *     *  *****                                                  -
CS   *     * ******                                                  -
```

```
CS    *
CS    *     ********(REPEAT FOR ALL BACK-GOING DIRECTIONS)
CS    *     *   COMPUTATIONAL COSINES                ALWAYS
CS    *     *   CELL FRONT-EDGE ANGULAR FLUX FOR FRONT PLANE
CS    *     *******
CS    *
CS    *     *******(REPEAT FOR ALL Z-INTERVALS)
CS    *               (INTERVAL 1 IS FIRST)
CS    *     *  ******(REPEAT FOR ALL BACK-GOING DIRECTIONS)
CS    *     *  * COMPUTATIONAL COSINES                ALWAYS
CS    *     *  * HORIZONTAL-EDGE ANGULAR FLUX         ALWAYS
CS    *     *  * VERTICAL-EDGE ANGULAR FLUX           ALWAYS
CS    *     *  * CELL BACK-EDGE ANGULAR FLUX          ALWAYS
CS    *     *  *****
CS    *     * ******
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 - RAFLXM - (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-------------------------------------------------------------------------
CR          SPECIFICATIONS      (1D RECORD)
C
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,EFFK,POWER
C
CW    8 =NUMBER OF WORDS
C
CD    NDIM               NUMBER OF DIMENSIONS
CD    NGROUP             NUMBER OF ENERGY GROUPS
CD    NINTI              NUMBER OF FIRST DIMENSION FINE MESH INTERVALS
CD    NINTJ              NUMBER OF SECOND DIMENSION FINE MESH INTERVALS
CD    NINTK              NUMBER OF THIRD DIMENSION FINE MESH INTERVALS.
CD                       NINTK.EQ.1 IF NDIM.LE.2
CD    EFFK               EFFECTIVE MULTIPLICATION FACTOR
CD    POWER              POWER IN WATTS TO WHICH FLUX IS NORMALIZED
C
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR          COMPUTATIONAL COSINES
C
CL    XMU,XETA,XI,WEIGHT,M,K
C
CW    6*MULT
C
CD    XMU            MU COSINE
CD    XETA           ETA COSINE
```

```
CD    XI              XI COSINE                                              -
CD    WEIGHT          WEIGHT                                                 -
CD    M               ANGLE INDEX IN THE OCTANT                              -
CD    K               Z-PLANE NUMBER                                         -
C                                                                            -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR           CELL FRONT-EDGE ANGULAR FLUX                                    -
C                                                                            -
CL    ((FBEDGE(I,J),I=NINTI),J=1,NINTJ)                                      -
C                                                                            -
CW    NINTI*NINTJ*MULT=NUMBER OF WORDS                                       -
C                                                                            -
CD    FBEDGE(I,J)     CELL FRONT-EDGE ANGULAR FLUX                           -
C                                                                            -
C                                                                            -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR           HORIZONTAL-EDGE ANGULAR FLUX                                    -
C                                                                            -
CL    ((HEDGE(I,J),I=1,NBDRYI),J=1,NINTJ)                                    -
C                                                                            -
CD    HEDGE(I,J)     HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX                   -
C                                                                            -
CW    NBDRYI*NINTJ*MULT=NUMBER OF WORDS                                      -
C                                                                            -
CD    NBDRYI          NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH           -
CD                    BOUNDARIES)                                            -
C                                                                            -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR           VERTICAL-EDGE ANGULAR FLUX                                      -
C                                                                            -
CL    ((VEDGE(I,J),I=1,NINTI),J=1,NBDRYJ)                                    -
C                                                                            -
CD    VEDGE(I,J)     VERTICAL-EDGE-BOUNDARY ANGULAR FLUX                     -
C                                                                            -
CW    NINTI*NBRDYJ*MULT=NUMBER OF WORDS                                      -
C                                                                            -
CD    NBDRYJ          NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH          -
CD                    BOUNDARIES)                                            -
C                                                                            -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR           CELL BACK-EDGE ANGULAR FLUX                                     -
C                                                                            -
CL    ((FBEDGE(I,J),I=NINTI),J=1,NINTJ)                                      -
C                                                                            -
CW    NINTI*NINTJ*MULT=NUMBER OF WORDS                                       -
C                                                                            -
CD    FBEDGE(I,J)     CELL BACK-EDGE ANGULAR FLUX                            -
C                                                                            -
C-------------------------------------------------------------------------
```



CEOF

RMFLUX

The RMFLUX code-dependent file contains, in binary form, the spherical harmonics regular angular flux moments for all spatial fine mesh points and all energy groups. It is optionally produced by the Solver Module.

```
C*****************************************************************************
C                        DATE 04/01/85                                        -
C                                                                             -
CF          RMFLUX-IV                                                         -
CE          REGULAR FLUX MOMENTS                                              -
C                                                                             -
C*****************************************************************************
C                                                                             -
CN               ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY            -
C                                                                             -
C-----------------------------------------------------------------------------
CS          FILE STRUCTURE                                                    -
CS                                                                            -
CS             RECORD TYPE                          PRESENT IF                -
CS             ============================       ==========                 -
CS             FILE IDENTIFICATION                  ALWAYS                    -
CS             SPECIFICATIONS                       ALWAYS                    -
CS                                                                            -
CS   *************(REPEAT FOR ALL GROUPS)                                    -
CS   *         REGULAR MOMENTS FLUXES               ALWAYS                    -
CS   *************                                                           -
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 - RMFLUX - (A6)                  -
CD    HNAME              HOLLERITH FILE NAME -        -(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          SPECIFICATIONS     (1D RECORD)                                    -
C                                                                             -
CL    NDIM,NGROUP,NINTI,NINTJ,NINTK,NORD,EFFK,POWER,OITNO                     -
C                                                                             -
CW    9=NUMBER OF WORDS                                                       -
C                                                                             -
CD    NDIM               NUMBER OF DIMENSIONS                                 -
CD    NGROUP             NUMBER OF GROUPS                                     -
CD    NINTI              NUMBER OF FIRST DIMENSION INTERVALS                  -
CD    NINTJ              NUMBER OF SECOND DIMENSION INTERVALS                 -
CD                       NINTJ.EQ.1 IF NDIM.EQ.1                              -
CD    NINTK              NUMBER OF THIRD DIMENSION INTERVALS                  -
```

```
CD                        NINTK.EQ.1 IF NDIM.LE.2                          -
CD     NORD               NUMBER OF LEGENDRE MOMENTS                       -
CD     EFFK               EFFECTIVE MULTIPLICATION FACTOR                  -
CD     POWER              POWER IN WATTS TO WHICH FLUX IS NORMALIZED       -
CD     OITNO              OUTER ITERATION NUMBER                           -
C-------------------------------------------------------------------------
C                                                                           -
C-------------------------------------------------------------------------
CR          REGULAR MOMENTS FLUXES ON MULTIDIMENSIONAL INTERVALS            -
C                              (2D RECORD)                                  -
C                                                                           -
CL     ((FLUX(M,I),M=1,NORD),I=1,NINTI)  ----NOTE STRUCTURE BELOW---        -
C                                                                           -
CW     NORD*NINTI=NUMBER OF WORDS                                           -
C                                                                           -
C      DO 1 K=1,NINTK                                                       -
C      DO 1 J=1,NINTJ                                                       -
C    1 READ (N) *LIST AS ABOVE*                                             -
C                                                                           -
CD     FLUX(M,I)          REGULAR FLUX MOMENTS ON FIRST DIMENSION          -
CD                        INTERVALS.                                       -
C                                                                           -
C-------------------------------------------------------------------------
CEOF                                                                        -
```

RZMFLX

The RZMFLX file is a binary, code-dependent file containing the spherical harmonics regular angular flux moments averaged over each zone for each energy group. The zones over which the fluxes are averaged are the zones used in the Solver Module and not the Edit Zones optionally used in the Edit Module.

```
C*****************************************************************************
C                        DATE 01/28/95                                        -
C                                                                             -
CF          RZMFLX-IV                                                         -
CE          REGULAR ZONE AVERAGED FLUX MOMENTS BY GROUP                       -
C                                                                             -
C*****************************************************************************
C                                                                             -
CN                ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY           -
C                                                                             -
C-----------------------------------------------------------------------------
CS          FILE STRUCTURE                                                    -
CS                                                                            -
CS             RECORD TYPE                          PRESENT IF                -
CS             ==============================       ==========                -
CS             FILE IDENTIFICATION                  ALWAYS                    -
CS             SPECIFICATIONS                       ALWAYS                    -
CS                                                                            -
CS   *************(REPEAT FOR ALL GROUPS)                                    -
CS   *         ZONE AVERAGED FLUX MOMENTS          ALWAYS                    -
CS   *************                                                           -
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 - RZMFLX - (A6)                  -
CD    HNAME              HOLLERITH FILE NAME -        -(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          SPECIFICATIONS     (1D RECORD)                                    -
C                                                                             -
CL    NDIM,NGROUP,NZONE,DUM,DUM,NORD,EFFK,POWER,OITNO                         -
C                                                                             -
CW    9=NUMBER OF WORDS                                                       -
C                                                                             -
CD    NDIM               NUMBER OF DIMENSIONS                                 -
CD    NGROUP             NUMBER OF GROUPS                                     -
CD    NZONE              NUMBER OF GEOMETRIC ZONES                            -
CD    DUM                DUMMY, NOT USED                                      -
CD    DUM                DUMMY, NOT USED                                      -
```

```
CD    NORD              NUMBER OF LEGENDRE MOMENTS                          -
CD    EFFK              EFFECTIVE MULTIPLICATION FACTOR                     -
CD    POWER             POWER IN WATTS TO WHICH FLUX IS NORMALIZED           -
CD    OITNO             OUTER ITERATION NUMBER                              -
C                                                                           -
C-------------------------------------------------------------------------
C                                                                           -
C-------------------------------------------------------------------------
CR          REGULAR FLUX MOMENTS AVERAGED OVER EACH ZONE                    -
C                              (2D RECORD)                                  -
C                                                                           -
CL    ((FLUX(M,I),M=1,NORD),I=1,NZONE)                                      -
C                                                                           -
CW    NORD*NZONE=NUMBER OF WORDS                                            -
C                                                                           -
C                                                                           -
CD    FLUX(M,I)         REGULAR FLUX MOMENT AVERAGES FOR EACH               -
CD                      ZONE.                                               -
C                                                                           -
C-------------------------------------------------------------------------
CEOF                                                                        -
```

SNXEDT

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 DANTSYS 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 05/12/83                               -
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                                                                      -
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.0          -
CS   *          SCATTERING MATRIX                   NORD.NE.0          -
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---------------------------------------------------------------------
CR          FILE CONTROL                                               -
C                                                                      -
```

```
CL    NGROUP,NISO,NORD,NED,IDPF,LNG,MAXUP,MAXDN,NPRIN,I2LP1
C
CW    10=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 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
CD    NPRIN           NUMBER OF PRINCIPAL CROSS SECTIONS (4 FOR SNXEDT)
CD    I2LP1           0/1 = NO/YES  2L+1 TERM WAS INCLUDED IN LIBRARY
C
C-------------------------------------------------------------------------
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)*MULT=NUMBER OF WORDS
C
CD    HISO(I)         HOLLERITH ISOTOPE LABEL FOR ISOTOPE (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+NPRIN
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
CN                    ALSO PRESENT WHEN NPRIN=5 IS:
C
CN                          HED(5) = 5HTRANS
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*MULT=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
CN                    ALSO PRESENT WHEN NPRIN=5 IS:
C
```

```
CN                        J=5  TRANSPORT CROSS SECTION                  -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          SCATTERING CONTROL BLOCK FOR GROUP N                        -
C                                                                       -
CC        PRESENT IF NORD.GT.0                                          -
C                                                                       -
CL    ((NGPB(L,J),L=1,NORD),J=1,NISO)                                   -
CL    ((IFSG(L,J),L=1,NORD),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                                                                       -
CC        PRESENT IF NORD.GT.0                                          -
C                                                                       -
CL    (SCAT(I),I=1,NTAB)                                                -
C                                                                       -
CW    NTAB*MULT=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 (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                                                                    -
```

SOLINP

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 01/28/95                                  -
C                                                                         -
CF          SOLINP                                                        -
CE          CODE DEPENDENT FILE OF INFORMATION SPECIFIC TO THE            -
CE          ONEDANT/TWODANT/THREEDANT SOLVER MODULES                      -
C                                                                         -
C                                                                         -
C*************************************************************************-
C-------------------------------------------------------------------------
CS          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=4 OR IBR=4        -
CS             BNDRY TRANSFER VECTOR LENGTHS        IBL=4 OR IBR=4        -
CS     ***********(REPEAT FOR ALL GROUPS)                                 -
CS     *       LEFT BDRY TRANSFER VECTOR            IBL=4                 -
CS     *       RIGHT BDRY TRANSFER VECTOR           IBR=4                 -
CS     ***********                                                        -
CS             FINE MESH DENSITY FACTORS            IDENX=2               -
CS             FINE MESH DENSITY VECTOR IN X        IDENX=1               -
CS             FINE MESH DENSITY VECTOR IN Y        IDENY=1               -
CS             FINE MESH DENSITY VECTOR IN Z        IDENZ=1               -
CS             MONTE CARLO OPTION QUANTITIES        MCOPT>0               -
CS             ADAPTIVE WEIGHTED DIAMOND PARAMS        ALWAYS             -
CS             RADIUS MODIFIERS IN X                IEVT=4.AND.IXM=1      -
CS             RADIUS MODIFIERS IN Y                IEVT=4.AND.IYM=1      -
CS             RADIUS MODIFIERS IN Z                IEVT=4.AND.IZM=1      -
CS             SN ORDER BY GROUP                    IGRPSN=1              -
CS             LEFT ALBEDOES                        IBEDOL.NE.0           -
CS             RIGHT ALBEDOES                       IBEDOR.NE.0           -
CS             BOTTOM ALBEDOES                      IBEDOB.NE.0           -
CS             TOP ALBEDOES                         IBEDOT.NE.0           -
CS             FRONT ALBEDOES                       IBEDOF.NE.0           -
CS             BACK ALBEDOES                        IBEDOK.NE.0           -
CS             SINGLE CHI ARRAY(FISSION SPECTRA)    INCHI=1               -
CS     ***********(REPEAT FOR ALL ZONES)                                  -
CS     *       CHI ARRAY(FISSION SPECTRA)           INCHI=2               -
CS     ***********                                                        -
CS             QUADRATURE WEIGHTS                   IQUAD=3               -
CS             QUADRATURE COSINES                   IQUAD=3               -
CS     ***********(REPEAT FOR ALL MOMENTS)                                -
CS     *       SOURCE SPECTRUM                      IQOPT=1               -
CS     *       SOURCE SPATIAL VECTOR(S)             IQOPT=2               -
CS     * *********(REPEAT FOR ALL GROUPS)                                 -
```

```
CS      * *      SOURCE POINTWISE FULL DISTRIBUTION IQOPT=3              -
CS      * *********                                                      -
CS      *        SOURCE SPECTRUM                      IQOPT=4           -
CS      *        SOURCE SPATIAL VECTOR(S)             IQOPT=4           -
                                                                         -
CS      *        SOURCE SPECTRUM                      IQOPT=6           -
CS      *        SOURCE POINTWISE FULL DISTRIBUTION IQOPT=6             -
CS      ***********                                                      -
CS      ***********(REPEAT FOR ALL GROUPS)                              -
CS      * *********(REPEAT FOR ALL FACES IN THE ORDER LEFT, RIGHT,      -
CS      * *          BOTTOM, TOP, FRONT, BACK, X=L,R,B,T,F,K BELOW)     -
CS      * *      BOUNDARY ISOTROPIC SOURCE            IQX=-1            -
CS      * *      BOUNDARY ANISOTROPIC SOURCE          IQX=+1            -
CS      * *      BOUNDARY SOURCE SPECTRUM             IQX=+2            -
CS      * *      BOUNDARY SOURCE SPACE VECTOR 1       IQX=+2            -
CS      * *      BOUNDARY SOURCE SPACE VECTOR 2       IQX=+2            -
CS      * *      BOUNDARY SOURCE ANGLE VECTOR         IQX=+2            -
CS      * *      BOUNDARY SOURCE SPECTRUM             IQX=+3            -
CS      * *      BOUNDARY SOURCE SPACE ARRAY          IQX=+3            -
CS      * *      BOUNDARY SOURCE ANGLE VECTOR         IQX=+3            -
CS      * *      BOUNDARY SOURCE SPECTRUM             IQX=+4            -
CS      * *      BOUNDARY SOURCE SPACE-ANGLE ARRAY    IQX=+4            -
CS      * *      BOUNDARY SOURCE SPECTRUM             IQX=+5            -
CS      * *      BOUNDARY SOURCE ANGLE-SPACE ARRAY    IQX=+5            -
CS      * *********                                                      -
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 - 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 DOUBLE PRECISION WORD            -
C                                                                        -
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.0                                           -
C                                                                        -
CL     (TITLE(I),I=1,12)                                                 -
C                                                                        -
CW     12=NUMBER OF A6 WORDS                                             -
C                                                                        -
C-------------------------------------------------------------------------
C                                                                        -
```

```
C-----------------------------------------------------------------------
CR          SPATIAL DIMENSION                                          -
C                                                                      -
CL    IDIMEN                                                           -
C                                                                      -
CW    1=NUMBER OF WORDS                                                -
C                                                                      -
CD    IDIMEN=1 FOR ONEDANT                                             -
C                                                                      -
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
CR          RAW CONTROLS AND DIMENSIONS                                -
C                                                                      -
CL    IEVT, ITH,   ISCT,  ISN,   IQUAD, ISTART,ICSM, INCHI,IBL, IBR,  -
CL  1 IDENX, IPVT,  I2ANG, IQOPT, IQAN,  IQL,   IQR,  OITM, IITL,ITM,  -
CL  2 ITLIM,I1,    FLUXP, XSECTP,FISSRP,SOURCP,GEOMP,IANG, IACC,IRMFLX,-
CL  3 IGRPSN, IAFLUX,ISBEDO,IBALP, DUM3,  IBB,   IBT,  IITLD,IQT, IQB, -
CL  4 IXM,  IYM,   IZM,   IDENY, IDENZ, ITMPMX, MFGACC, JUPDCH,        -
CL  5 JACC,  IBF,  IBK, MCNHIS, MCNHST, MCNTR, MCITS, MCOPT, MCNRS,    -
CL  6 MCPRNT, MCSEED, IMTSRC, IMBSLA, IMBSRA, IMBSBA, IMBSTA, IQF, IQK,-
CL  7 IBEDOL, IBEDOR, IBEDOB, IBEDOT, IBEDOF, IBEDOK, NOSIGF, IPLANT,  -
CL  8 JSRITE, JSBOTT, JSTOP,  JSLEFT, EIGONLY, FCNRAY, FCNTR, NODAL    -
CL  9 NPROC, AVATAR                                                    -
CL  8 (DUM5(I),I=1,22)                                                 -
C                                                                      -
CW    100=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/ZONEWISE 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  IDENX    0/1/2 - NONE/FINE MESH DENSITY FACTORS BY XMESH/       -
CD                       FINE MESH DENSITY FACTORS FOR EVERY MESH      -
CD  12  IPVT     0/1/2 - NONE/K-EFF/ALPHA  PARAMETRIC EIGENVALUE TYPE  -
CD  13  I2ANG    0/1 - NO/YES  DO 2 ANGLE SLAB CALCULATION             -
CD  14  IQOPT    0/1/2/3/4/5/6 - INHOMOGENEOUS SOURCE OPTION           -
CD  15  IQAN     INHOMOGENEOUS SOURCE LEGENDRE ORDER                   -
CD  16  IQL      -1/0/1/2/3/4/5  LEFT BOUNDARY SOURCE OPTION           -
CD  17  IQR      -1/0/1/2/3/4/5  RIGHT BOUNDARY SOURCE OPTION          -
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  IRMFLX   WRITE CODE-DEPENDENT ZONE FLUXES                      -
CD                                                                     -
CD  31  IGRPSN   GROUP DEP SN ORDERS (GRPSN) READ IN                   -
CD  32  RAFLUX   0/1 - NO/YES  WRITE ANGULAR FLUX FILE RAFLUX          -
CD  33  ISBEDO   ALBEDO OPTION                                         -
CD  34  IBALP    0/1 - NO/YES  PRINT BALANCES BY ZONE                  -
```

```
CD  35  DUM3     NOT USED                                                -
CD  36  IBB      0/1/2/3/4 - BOTTOM BDRY CONDITION                       -
CD  37  IBT      0/1/2/3/4 - TOP BDRY CONDITION                          -
CD  38  IITLD    TIMELIMIT IN SECONDS                                    -
CD  39  IQT      -1/0/1/2/3/4/5  TOP BOUNDARY SOURCE OPTION              -
CD  40  IQB      -1/0/1/2/3/4/5  BOTTOM BOUNDARY SOURCE OPTION           -
CD                                                                       -
CD  41  IXM      0/1 - NO/YES RADIAL MODIFIERS FOR X                     -
CD  42  IYM      0/1 - NO/YES RADIAL MODIFIERS FOR Y                     -
CD  43  IZM      0/1 - NO/YES RADIAL MODIFIERS FOR Z                     -
CD  44  IDENY    0/1 - NO/FINE MESH DENSITY FACTORS BY YMESH             -
CD  45  IDENZ    0/1 - NO/FINE MESH DENSITY FACTORS BY YMESH             -
CD  46  ITMPMX   NOT USED (RESERVED)                                     -
CD  47  MFGACC   NOT USED (RESERVED)                                     -
CD  48  JUPDCH   NOT USED (RESERVED)                                     -
CD  49  JACC     NOT USED (RESERVED)                                     -
CD  50  IBF      0/1/2/3/4 - FRONT BDRY CONDITION                        -
CD                                                                       -
CD  51  IBK      0/1/2/3/4 - BACK BDRY CONDITION                         -
CD  52  MCNHIS   NUMBER OF HISTORIES FOR MC OPTION                       -
CD  53  MCNHST   NUMBER OF HISTORIES FOR MC OPTION                       -
CD  54  MCNTR    NUMBER OF BATCHES FOR MC OPTION                         -
CD  55  MCITS    MAX NUMBER OF SUPEROUTERS IN MC OPTION                  -
CD  56  MCOPT    0/1 YES/NO TURN ON MC OPTION                            -
CD  57  MCNRS                                                            -
CD  58  MCPRNT   PRINT LEVEL FOR MC OPTION                               -
CD  59  MCSEED   REST INITIAL RANDOM SEED                                -
CD                                                                       -
CD  60  IMTSRC   USED IN MC OPTION                                       -
CD  61  IMBLSA   USED IN MC OPTION                                       -
CD  62  IMBSRA   USED IN MC OPTION                                       -
CD  63  IMBSBA   USED IN MC OPTION                                       -
CD  64  IMBSTA   USED IN MC OPTION                                       -
CD                                                                       -
CD  65  IQF      -1/0/1/2/3/4/5  FRONT BOUNDARY SOURCE OPTION           -
CD  66  IQK      -1/0/1/2/3/4/5  BACK BOUNDARY SOURCE OPTION            -
CD                                                                       -
CD  67  IBEDOL   0/1/2 - NONE/LEFT ALBEDO/ALBEDO SPATIAL DISTRIBUTION   -
CD  68  IBEDOR   0/1/2 - NONE/RIGHT ALBEDO/ALBEDO SPATIAL DISTRIBUTION  -
CD  69  IBEDOB   0/1/2 - NONE/BOTTOM ALBEDO/ALBEDO SPATIAL DISTRIBUTION-
CD  70  IBEDOT   0/1/2 - NONE/TOP ALBEDO/ALBEDO SPATIAL DISTRIBUTION    -
CD  71  IBEDOF   0/1/2 - NONE/FRONT ALBEDO/ALBEDO SPATIAL DISTRIBUTION  -
CD  72  IBEDOK   0/1/2 - NONE/BACK ALBEDO/ALBEDO SPATIAL DISTRIBUTION   -
CD  73  NOSIGF   0/1   -  NONE/SET NUSGF ZERO FOR SOURCE PROBLEMS       -
CD  73  IPLANT   0/1   -  NONE/PLANET VARIABLE PRESENT                  -
CD  75  JSRITE   0/i = NO/YES WRITE RIGHT BOUNDARY FLUX AT i            -
CD  76  JSBOTT   0/i = NO/YES WRITE BOTTOM BOUNDARY FLUX AT i           -
CD  77  JSTOP    0/i = NO/YES WRITE TOP BOUNDARY FLUX AT i               -
CD  78  JSLEFT   0/i = NO/YES WRITE LEFT BOUNDARY FLUX AT i             -
                                                                         -
CD  79  EIGONLY  0/1 NO/YES ONLY CONVERGE EIGENVALUE AND FISSIONS        -
CD  80-85        (RESERVED FOR FUTURE USE)                               -
CD  86  FCNRAY   NO. OF RAY TRACINGS/BATCH (RAY TRACE OPTION)            -
CD  87  FCNTR    NUMBER OF BATCHES IN THE RAY TRACE OPTION               -
CD  88  NODAL    0/1/2 DD OR AWDD/CL/LL NODAL SPATIAL DIFFERENCING       -
CD  89  NPROC    NUMBER OF PROCESSORS TO BE USED IN PVM VERSION          -
CD  90-92        (RESERVED FOR TIME DEPENDENCE)                          -
CD  93  AVATAR   0/1 NO/YES WRITE THE AVATAR FILE                        -
C---------------------------------------------------------------------------
C                                                                        -
C---------------------------------------------------------------------------
CR          RAW FLOATING DATA                                            -
C                                                                        -
CL    EV,   NORM, EPSO, EPSI, BHGT, BWTH, EVM, PV, XLAL, XLAH,           -
CL  1 XLAX, POD,  EPSR, EPSX, EPST,                                      -
```

```
CL  2 (DUM(I),I=1,10),                                                  -
CL  3 EFACT, TO, TS, EOM, SIGTH, TRCOR, PLANET, FCSRC, XMCSB, XMCBLT,   -
CL  4 (DUM1(I),I=1,55),                                                 -
CL  5 (EXTRAS(I),I=1,110)                                               -
C                                                                       -
CW    200*MULT=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  POD       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                                   -
CD                                                                      -
CD  16  DUM       VECTOR NOT USED                                       -
CD                                                                      -
CD  26  EFACT     NOT USED (RESERVED FOR TIME DEPENDENCE)               -
CD  27  TO        NOT USED (RESERVED FOR TIME DEPENDENCE)               -
CD  28  TS        NOT USED (RESERVED FOR TIME DEPENDENCE)               -
CD  29  EOM       NOT USED (RESERVED FOR TIME DEPENDENCE)               -
CD  30  SIGTH     NOT USED (RESERVED FOR TIME DEPENDENCE)               -
CD                                                                      -
CD  31  TRCOR     TRANSPORT CORRECTION INDICATOR                        -
CD  32  PLANET    PLANET INDICATOR                                      -
CD  33  FCSRC     USE FIRST COLLISION SOURCE OPTION                     -
CD  34  XMCSB     BIASING PARAMETER IN MC OPTION                        -
CD  35  XMCBLT    BOUNDARY LAYER THICKNESS IN MC OPTION                 -
CD                                                                      -
CD  36  FCWCO     WEIGHT CUTOFF FOR FIRST COLLISION RAYS                -
CD                                                                      -
CD  91  EXTRAS    VECTOR USED BY INDIVIDUAL SOLVERS                     -
C                                                                       -
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-----------------------------------------------------------------------
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-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          BOUNDARY TRANSFER FIRST SOURCE GROUP                        -
C                                                                       -
CC      PRESENT IF IBL.EQ.4 .OR. IBR.EQ.4                               -
C                                                                       -
```

```
CL     (IFSGL(N),N=1,NGROUP),(IFSGR(N),N=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       NGROUP    NUMBER OF ENERGY GROUPS                              -
C                                                                       -
C-----------------------------------------------------------------------
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-----------------------------------------------------------------------
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 IFSGL(N)  -
CD                 INTO GROUP N, NEXT IS FROM GROUP IFSGL(N)-1 TO N, ETC.-
CD       NW        LENL(N)                                              -
CD       N         NUMBER OF THE RECEIVING GROUP                        -
C                                                                       -
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                                                                       -
C-----------------------------------------------------------------------
CR           FINE MESH DENSITY FACTORS                                  -
C                                                                       -
CC      PRESENT IF IDENX.EQ.2                                           -
C                                                                       -
```

```
CL     (DEN(I),I=1,IT)----NOTE STRUCTURE BELOW----
C
CW     IT*MULT=NUMBER OF WORDS
C
CS     DO 1 K=1,KT
CS     DO 1 J=1,JT
CS   1 READ (N) *LIST AS ABOVE*
C
CD       IT        NUMBER OF FINE MESH INTERVALS
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR            FINE MESH DENSITY VECTOR IN X
C
CC      PRESENT IF IDENX.EQ.1
C
CL     (DEN(I),I=1,IT)
C
CW     IT*MULT=NUMBER OF WORDS
C
CD       IT        NUMBER OF FINE MESH INTERVALS IN X DIRECTION
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR            FINE MESH DENSITY VECTOR IN Y
C
CC      PRESENT IF IDENY.EQ.1
C
CL     (DEN(J),J=1,JT)
C
CW     JT*MULT=NUMBER OF WORDS
C
CD       JT        NUMBER OF FINE MESH INTERVALS IN Y DIRECTION
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR            FINE MESH DENSITY VECTOR IN Z
C
CC      PRESENT IF IDENZ.EQ.1
C
CL     (DEN(K),K=1,KT)
C
CW     KT*MULT=NUMBER OF WORDS
C
CD       KT        NUMBER OF FINE MESH INTERVALS IN Z DIRECTION
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR            MONTE CARLO OPTION PARAMETERS
C
CC      PRESENT IF MCOPT.EQ.1
C
CL     (JBND(I),I=1,2)
C
CW     2*MULT=NUMBER OF WORDS
C
CL     (IBND(I),I=1,2)
C
CW     2*MULT=NUMBER OF WORDS
C
CL     (MCREG(G),G=1,IGM)
```

```
C                                                                       -
CW     IGM*MULT=NUMBER OF WORDS                                         -
C                                                                       -
CD       IGM       NUMBER OF ENERGY GROUPS                            -
C                                                                       -
CL     (MCTSRC(I),I=1,12)                                               -
C                                                                       -
CW     12*MULT=NUMBER OF WORDS                                          -
C                                                                       -
CL     (EGIM(G),G=1,IGM)                                                -
C                                                                       -
CW     IGM*MULT=NUMBER OF WORDS                                         -
C                                                                       -
CD       IGM       NUMBER OF ENERGY GROUPS                            -
C                                                                       -
CL     (MIMP(K),K=1,IMJM)                                               -
C                                                                       -
CW     IMJM*MULT=NUMBER OF WORDS                                        -
C                                                                       -
CD       IMJM     NUMBER OF COARSE MESH INTERVALS                      -
C                                                                       -
CL     (VIMP(K),K=1,ISN/2)                                              -
C                                                                       -
CW     ISN/2*MULT=NUMBER OF WORDS                                       -
C                                                                       -
CD       ISN     SN ORDER                                               -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR           AWDD PARAMETERS                                            -
C                                                                       -
CC        PRESENT ALWAYS                                                -
C                                                                       -
CL     (WDAMP(G),G=1,IGM)                                               -
C                                                                       -
CW     IGM*MULT=NUMBER OF WORDS                                         -
C                                                                       -
CD       IGM       NUMBER OF ENERGY GROUPS                            -
C                                                                       -
CL     (THRSHD(G),G=1,IGM)                                              -
C                                                                       -
CW     IGM*MULT=NUMBER OF WORDS                                         -
C                                                                       -
CD       IGM       NUMBER OF ENERGY GROUPS                            -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR           RADIAL MODIFIER IN X                                       -
C                                                                       -
CC        PRESENT IF IEVT.EQ.4 .AND. IXM.EQ.1                           -
C                                                                       -
CL     (RM(I),I=1,IM)                                                   -
C                                                                       -
CW     IM*MULT=NUMBER OF WORDS                                          -
C                                                                       -
CD       IM        NUMBER OF COARSE MESH INTERVALS                     -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR           RADIAL MODIFIER IN Y                                       -
C                                                                       -
CC        PRESENT IF IEVT.EQ.4 .AND. IYM.EQ.1                           -
```

```
C
CL     (RM(I),J=1,JM)
C
CW     JM*MULT=NUMBER OF WORDS
C
CD       JM        NUMBER OF COARSE MESH INTERVALS
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           RADIAL MODIFIER IN Z
C
CC         PRESENT IF IEVT.EQ.4 .AND. IZM.EQ.1
C
CL     (RM(K),K=1,KM)
C
CW     KM*MULT=NUMBER OF WORDS
C
CD       KM        NUMBER OF COARSE MESH INTERVALS
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           SN ORDER BY GROUP
C
CC         PRESENT IF IGRPSN=1
C
CL     (GRPSN(IG),IG=1,IGM)
C
CW     IGM*MULT=NUMBER OF WORDS
C
CD       IGM        NUMBER OF ENERGY GROUPS
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           LEFT ALBEDOES
C
CC       PRESENT IF IBEDOL.EQ.1
C
CL     (LBEDO(N),N=1,NGROUP)
C
CW     NGROUP*MULT=NUMBER OF WORDS
C
C    ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---
C
CC       PRESENT IF IBEDOL.EQ.2
C
CL     (LBEDO(N),N=1,NGROUP)----NOTE STRUCTURE BELOW----
C
CW     NGROUP*MULT=NUMBER OF WORDS
C
CS     DO 1 K=1,KT
CS     DO 1 J=1,JT
CS   1 READ(N) *LIST AS ABOVE*
C
C-------------------------------------------------------------------
C
C-------------------------------------------------------------------
CR           RIGHT ALBEDOES
C
CC       PRESENT IF IBEDOR.EQ.1
C
CL     (RBEDO(N),N=1,NGROUP)
C
```

```
CW    NGROUP*MULT=NUMBER OF WORDS
C
C   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---
C
CC      PRESENT IF IBEDOR.EQ.2
C
CL    (RBEDO(N),N=1,NGROUP)----NOTE STRUCTURE BELOW----
C
CW    NGROUP*MULT=NUMBER OF WORDS
C
CS    DO 1 K=1,KT
CS    DO 1 J=1,JT
CS  1 READ(N) *LIST AS ABOVE*
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR          BOTTOM ALBEDOES
C
CC      PRESENT IF IBEDOB.EQ.1
C
CL    (BBEDO(N),N=1,NGROUP)
C
CW    NGROUP*MULT=NUMBER OF WORDS
C
C   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---
C
CC      PRESENT IF IBEDOB.EQ.2
C
CL    (BBEDO(N),N=1,NGROUP)----NOTE STRUCTURE BELOW----
C
CW    NGROUP*MULT=NUMBER OF WORDS
C
CS    DO 1 K=1,KT
CS    DO 1 I=1,IT
CS  1 READ(N) *LIST AS ABOVE*
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR          TOP ALBEDOES
C
CC      PRESENT IF IBEDOT.EQ.1
C
CL    (TBEDO(N),N=1,NGROUP)
C
CW    NGROUP*MULT=NUMBER OF WORDS
C
C   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---
C
CC      PRESENT IF IBEDOT.EQ.2
C
CL    (TBEDO(N),N=1,NGROUP)----NOTE STRUCTURE BELOW----
C
CW    NGROUP*MULT=NUMBER OF WORDS
C
CS    DO 1 K=1,KT
CS    DO 1 I=1,IT
CS  1 READ(N) *LIST AS ABOVE*
C
C-------------------------------------------------------------------------
C
C-------------------------------------------------------------------------
CR          FRONT ALBEDOES
C
```

```
CC      PRESENT IF IBEDOF.EQ.1                                          -
C                                                                       -
CL    (FBEDO(N),N=1,NGROUP)                                             -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
C   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---     -
C                                                                       -
CC      PRESENT IF IBEDOF.EQ.2                                          -
C                                                                       -
CL    (FBEDO(N),N=1,NGROUP)----NOTE STRUCTURE BELOW----                 -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
CS    DO 1 J=1,JT                                                       -
CS    DO 1 I=1,IT                                                       -
CS  1 READ(N) *LIST AS ABOVE*                                           -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          BACK ALBEDOES                                               -
C                                                                       -
CC      PRESENT IF IBEDOK.EQ.1                                          -
C                                                                       -
CL    (BKBEDO(N),N=1,NGROUP)                                            -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
C   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---     -
C                                                                       -
CC      PRESENT IF IBEDOK.EQ.2                                          -
C                                                                       -
CL    (BKBEDO(N),N=1,NGROUP)----NOTE STRUCTURE BELOW----                -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
CS    DO 1 J=1,JT                                                       -
CS    DO 1 I=1,IT                                                       -
CS  1 READ(N) *LIST AS ABOVE*                                           -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          CHI - FISSION SPECTRA                                       -
C                                                                       -
CC      PRESENT IF INCHI.NE.0                                           -
C                                                                       -
CL    (CHI(N),N=1,NGROUP)                                               -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          QUADRATURE WEIGHTS                                          -
C                                                                       -
CC      PRESENT IF IQUAD.EQ.3                                           -
C                                                                       -
CL    (WGT(M),M=1,MM)                                                   -
C                                                                       -
CW    MM*MULT=NUMBER OF WORDS                                           -
C                                                                       -
CD      MM        NUMBER OF ANGLES IN QUADRATURE SET                    -
C                                                                       -
```

```
C-------------------------------------------------------------------------
C                                                                        -
C-------------------------------------------------------------------------
CR          QUADRATURE COSINES                                           -
C                                                                        -
CC      PRESENT IF IQUAD.EQ.3                                            -
C                                                                        -
CL    (MU(M),M=1,MM)                                                     -
C                                                                        -
CW    MM*MULT=NUMBER OF WORDS                                            -
C                                                                        -
C-------------------------------------------------------------------------
C                                                                        -
C-------------------------------------------------------------------------
CR          SOURCE SPECTRUM                                              -
C                                                                        -
CC      PRESENT IF IQOPT.EQ.1 .OR. IQOPT.EQ.4 .OR. IQOPT.EQ.6            -
C                                                                        -
CL    (SOURCE(N),N=1,NGROUP)                                             -
C                                                                        -
CW    NGROUP*MULT=NUMBER OF WORDS                                        -
C                                                                        -
C-------------------------------------------------------------------------
C                                                                        -
C-------------------------------------------------------------------------
CR          SOURCE SPATIAL VECTOR IN FIRST DIMENSION                     -
C                                                                        -
CC      PRESENT IF IQOPT.EQ.2 .OR. IQOPT.EQ.4                            -
C                                                                        -
CL    (SOURCX(I),I=1,IT)                                                 -
C                                                                        -
CW    IT*MULT=NUMBER OF WORDS                                            -
C                                                                        -
C-------------------------------------------------------------------------
C                                                                        -
C-------------------------------------------------------------------------
CR          SOURCE SPATIAL VECTOR IN SECOND DIMENSION                    -
C                                                                        -
CC      PRESENT IF IQOPT.EQ.2 .OR. IQOPT.EQ.4 .AND. IDIMEN.GE.2          -
C                                                                        -
CL    (SOURCY(J),J=1,JT)                                                 -
C                                                                        -
CW    JT*MULT=NUMBER OF WORDS                                            -
C                                                                        -
C-------------------------------------------------------------------------
C                                                                        -
C-------------------------------------------------------------------------
CR          SOURCE SPATIAL VECTOR IN THIRD DIMENSION                     -
C                                                                        -
CC      PRESENT IF IQOPT.EQ.2 .OR. IQOPT.EQ.4 .AND. IDIMEN.EQ.3          -
C                                                                        -
CL    (SOURCZ(K),K=1,KT)                                                 -
C                                                                        -
CW    KT*MULT=NUMBER OF WORDS                                            -
C                                                                        -
C-------------------------------------------------------------------------
C                                                                        -
C-------------------------------------------------------------------------
CR          SOURCE POINTWISE FULL DISTRIBUTION                           -
C                                                                        -
CC      PRESENT IF IQOPT.EQ.3 .OR. IQOPT.EQ.6                            -
C                                                                        -
CL    (SOURCX(I),I=1,IT)----NOTE STRUCTURE BELOW----                     -
C                                                                        -
CW    IT*MULT=NUMBER OF WORDS                                            -
```

```
C                                                                       -
CS    DO 1 K=1,KT                                                       -
CS    DO 1 J=1,JT                                                       -
CS  1 READ(N) *LIST AS ABOVE*                                           -
C                                                                       -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          BOUNDARY ISOTROPIC SOURCE (LEFT SHOWN)                      -
C                                                                       -
CC      PRESENT IF IBL.EQ.-1 .AND. IDIMEN.EQ.1 .AND. GROUP.EQ.1         -
C                                                                       -
CL    (SILEFT(N),N=1,NGROUP)                                            -
C                                                                       -
CW    NGROUP*MULT=NUMBER OF WORDS                                       -
C                                                                       -
CD      SILEFT(N)         LEFT BOUNDARY SOURCE FOR GROUP N              -
C                                                                       -
C   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---   ---     -
C                                                                       -
CC      PRESENT IF IBL.EQ.-1 .AND. IDIMEN.GT.1                          -
C                                                                       -
CL    (SILEFT(J),J=1,JT)----NOTE STRUCTURE BELOW----                    -
C                                                                       -
CW    JT*MULT=NUMBER OF WORDS                                           -
C                                                                       -
CS    DO 1 K=1,KT                                                       -
CS  1 READ(N) *LIST AS ABOVE*                                           -
C                                                                       -
CD      SILEFT(J)         LEFT BOUNDARY SOURCE FOR THE GROUP AT MESH    -
CD                        INTERVAL (J,K)                                -
CD      JT                NUMBER OF FINE MESH INTERVALS IN THE SECOND   -
CD                        DIMENSION. EQUALS 1 FOR ONEDANT.              -
CD      KT                NUMBER OF FINE MESH INTERVALS IN THE THIRD    -
CD                        DIMENSION. EQUALS 1 FOR ONEDANT AND TWODANT.  -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          BOUNDARY ANISOTROPIC SOURCE (LEFT SHOWN)                    -
C                                                                       -
CC      PRESENT IF IBL.EQ.+1                                            -
C                                                                       -
CL    (SALEFT(M),M=1,MMHALF)----NOTE STRUCTURE BELOW----                -
C                                                                       -
CW    MMHALF*MULT=NUMBER OF WORDS                                       -
C                                                                       -
CS    DO 1 K=1,KT                                                       -
CS    DO 1 J=1,JT                                                       -
CS  1 READ(N) *LIST AS ABOVE*                                           -
C                                                                       -
CD      SALEFT(M)       LEFT BOUNDARY ANGULAR SOURCE DISTRIBUTION FOR   -
CD                      THE GROUP AT INTERVAL (J,K)                     -
CD      MMHALF          NUMBER OF INCOMING ANGLES IN THE SOURCE         -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
C-----------------------------------------------------------------------
CR          BOUNDARY SOURCE SPECTRUM (LEFT SHOWN)                       -
C                                                                       -
CC      PRESENT IF IQL.GE.2 .AND. IQL.LE.4 .AND. GROUP.EQ.1             -
C                                                                       -
CL    (SOURCE(N),N=1,NGROUP)                                            -
C                                                                       -
```

```
CW    NGROUP*MULT=NUMBER OF WORDS                                          -
C                                                                          -
C--------------------------------------------------------------------------
C                                                                          -
C--------------------------------------------------------------------------
CR          BOUNDARY SOURCE SPATIAL DISTRIBUTION VECTOR 1 (LEFT SHOWN)     -
C                                                                          -
CC      PRESENT IF IQL.EQ.2 .AND. GROUP.EQ.1                               -
C                                                                          -
CL    (SOURCY(J),J=1,JT)                                                   -
C                                                                          -
CW    JT*MULT=NUMBER OF WORDS                                              -
C                                                                          -
C--------------------------------------------------------------------------
C                                                                          -
C--------------------------------------------------------------------------
CR          BOUNDARY SOURCE SPATIAL DISTRIBUTION VECTOR 2 (LEFT SHOWN)     -
C                                                                          -
CC      PRESENT IF IQL.EQ.2 .AND. GROUP.EQ.1                               -
C                                                                          -
CL    (SOURCZ(K),K=1,KT)                                                   -
C                                                                          -
CW    KT*MULT=NUMBER OF WORDS                                              -
C                                                                          -
C--------------------------------------------------------------------------
C                                                                          -
C--------------------------------------------------------------------------
CR          BOUNDARY SOURCE SPACE ARRAY (LEFT SHOWN)                       -
C                                                                          -
CC      PRESENT IF IQL.EQ.3 .AND. GROUP.EQ.1                               -
C                                                                          -
CL    (SOURCY(J),J=1,JT)----NOTE STRUCTURE BELOW----                       -
C                                                                          -
CW    JT*MULT=NUMBER OF WORDS                                              -
C                                                                          -
CS    DO 1 K=1,KT                                                          -
CS  1 READ(N) *LIST AS ABOVE*                                              -
C                                                                          -
C--------------------------------------------------------------------------
C                                                                          -
C--------------------------------------------------------------------------
CR          BOUNDARY SOURCE SPACE-ANGLE ARRAY (LEFT SHOWN)                 -
C                                                                          -
CC      PRESENT IF IQL.EQ.4 .AND. GROUP.EQ.1                               -
C                                                                          -
CL    (SOURCY(J),J=1,JT)----NOTE STRUCTURE BELOW----                       -
C                                                                          -
CW    JT*MULT=NUMBER OF WORDS                                              -
C                                                                          -
CS    DO 1 M=1,MMHALF                                                      -
CS    DO 1 K=1,KT                                                          -
CS  1 READ(N) *LIST AS ABOVE*                                              -
C                                                                          -
C--------------------------------------------------------------------------
C                                                                          -
C--------------------------------------------------------------------------
CR          BOUNDARY SOURCE ANGLE-SPACE ARRAY (LEFT SHOWN)                 -
C                                                                          -
CC      PRESENT IF IQL.EQ.5 .AND. GROUP.EQ.1                               -
C                                                                          -
CL    (SALEFT(M),M=1,MMHALF)----NOTE STRUCTURE BELOW----                   -
C                                                                          -
CW    MMHALF*MULT=NUMBER OF WORDS                                          -
C                                                                          -
CS    DO 1 K=1,KT                                                          -
```

```
CS     DO 1 J=1,JT                                                      -
CS   1 READ(N) *LIST AS ABOVE*                                          -
C                                                                       -
C-----------------------------------------------------------------------
C                                                                       -
CEOF                                                                    -
```

UCFLUX

The UCFLUX file is a binary, code-dependent interface file containing the necessary information for a restart calculation when using the ray tracing first collision option in the TWODANT Solver Module.

```
C*****************************************************************-
C                              DATE 02/09/95                      -
C                                                                 -
CF              UCFLUX                                            -
CE              CODE DEPENDENT UNCOLLIDED FLUX RESTART FILE       -
C                                                                 -
C*****************************************************************-

C-----------------------------------------------------------------
C                                                                 -
C                                                                 -
CN      THIS FILE ALLOWS A RESTART WHEN USING THE FCSRC OPTION    -
C                                                                 -
C                                                                 -
C-----------------------------------------------------------------

C-----------------------------------------------------------------
CS              FILE STRUCTURE                                    -
CS                                                                -
CS              RECORD TYPE                    PRESENT IF         -
CS              ===========================    ==========         -
CS              INT SPECIFICATIONS             ALWAYS             -
CS              FP SPECIFICATIONS              ALWAYS             -
CS              NUMBER OF TRACKS               I2.GT.0            -
CS              INTEGRAL FLUXES                I2.GT.0            -
CS                                                                -
CS   ***************(REPEAT FOR NSGRP GROUPS)                     -
CS   *          UNCOLLIDED FLUX MOMENTS        ALWAYS             -
CS   ***************                                              -
C                                                                 -
CC              NSGRP IS THE NUMBER OF GROUPS WITH AN INHOMOGENEOUS -
CC              SOURCE                                            -
C                                                                 -
C-----------------------------------------------------------------

C-----------------------------------------------------------------
CR              INT SPECIFICATIONS             (1D RECORD)        -
C                                                                 -
CL      NHSTOT,NXYSCORE,IPREVTR,NWDS                              -
C                                                                 -
CW      4 =NUMBER OF WORDS                                        -
C                                                                 -
CD      NHSTOT          NUMBER OF RAYS TRACED                     -
CD      NXYSCORE        NUMBER OF RAYS TERMINATED                 -
CD      IPREVTR         NUMBER OF PREVIOUS TRIALS                 -
CD      NWDS            TOTAL NUMBER OF WORDS IN FILE             -
C                                                                 -
C-----------------------------------------------------------------
```

```
C-------------------------------------------------------------------------
CR              FP SPECIFICATIONS                (1D RECORD)             -
C                                                                        -
CL      WTMIN,WTMAX,WTTOT,PTIME                                          -
C                                                                        -
CW      4 =NUMBER OF WORDS                                               -
C                                                                        -
CD      WTMIN           MINIMUM RAY WEIGHT                               -
CD      WTMAX           MAXIMUM RAY WEIGHT                               -
CD      WTTOT           TOTAL RAY WEIGHT                                 -
CD      PTIME           CPU TIME                                         -
C                                                                        -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR              NUMBER OF TRACKS         (2D RECORD)                     -
C                                                                        -
CC              PRESENT IF I2.GT.0                                       -
C                                                                        -
CL      ((NTENTR(I,J),I=1,IT),J=1,JT)                                    -
C                                                                        -
CW      IT*JT =NUMBER OF WORDS                                           -
C                                                                        -
CD      NTENTR(I,J)             NUMBER OF RAYS ENTERING EACH FINE MESH   -
C                                                                        -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR              INTEGRAL FLUXES          (2D RECORD)                     -
C                                                                        -
CC              PRESENT IF I2.GT.0                                       -
C                                                                        -
CL      ((FLRT(I,J),I=1,IT),J=1,JT), ((FLRT2(I,J),I=1,IT),J=1,JT)        -
C                                                                        -
CW      IT*JT*2 =NUMBER OF WORDS                                         -
C                                                                        -
CD      FLRT(I,J)               INTEGRAL FLUXES FOR VARIANCE CALC        -
CD      FLR2T(I,J)              INTEGRAL FLUXES**2 FOR VARIANCE CALC     -
C                                                                        -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR              LEAKAGES                 (1D RECORD)                     -
C                                                                        -
CL      (FCRL(I),I=1,IGM),(FCLL(I),I=1,IGM),(FCTL(I),I=1,IGM),           -
CL      (FCBL(I),I=1,IGM)                                                -
C                                                                        -
CW      IGM*4 =NUMBER OF WORDS                                           -
C                                                                        -
CD      FCRL(I)                 FCSRC RIGHT LEAKAGE (UNNORMALIZED)       -
CD      FCLL(I)                 FCSRC LEFT LEAKAGE (UNNORMALIZED)        -
CD      FCTL(I)                 FCSRC TOP LEAKAGE (UNNORMALIZED)         -
CD      FCBL(I)                 FCSRC BOTTOM LEAKAGE (UNNORMALIZED)      -
C                                                                        -
C-------------------------------------------------------------------------

C-------------------------------------------------------------------------
CR              UNCOLLIDED FLUX MOMENTS (3D RECORD)                      -
C                                                                        -
CL      (((FLUC(I,J,N),I=1,IT),J=1,JT),N=1,NM)                           -
C                                                                        -
```

```
CW      IT*JT*NM =NUMBER OF WORDS                                       -
C                                                                       -
CD      FLUC(I,J,N)             UNCOLLIDED FLUX MOMENTS MULTIPLIED BY   -
CD                              VOLUME, UNNORMALIZED                    -
C                                                                       -
C-----------------------------------------------------------------------
```

OVERWRITTEN INPUT FILES

On occasion, an input file has the same name as a normal output file. For instance, the RTFLUX file may be input, but a file named RTFLUX containing the fluxes is always written by any SOLVER at the completion of the run.

Such a file will be overwritten without comment if it is in the local filespace. DANTSYS only writes to the local filespace. For a list of the files that DANTSYS might write, the user is referred to Table 13.1, "Files Read and Written," on page 13-9 in chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure".

The user can protect input cross-section files through use of a search path name. See any of the User's Guides for a discussion of how to do this.

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).

CODE ABSTRACTS

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

16

XTM — Transport Methods Group

Los Alamos
National Laboratory

INTRODUCTION

This chapter contains abstracts for each of the solvers in the DANTSYS package.

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

by
Ray E. Alcouffe, Forrest W. Brinkley,
Duane R. Marr, and R. Douglas O'Dell

ONEDANT ABSTRACT

1. Program Identification: ONEDANT

2. Computer for which Program is designed:

   The current release is designed for Unix-like systems. The specific computers supported fall into two categories: Long word computers and short word computers. The program has been implemented on the long word Cray-YMP and Cray 2 computers. It has also been implemented on the Sun, SGI, HP9000, and IBM RS6000 short word workstations. The workstation versions use double precision arithmetic. Older versions have run on CDC Cyber 205, Vax, and main frame IBM but are not supported in this release.

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, multigroup 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. Originally designed for the CDC-7600 computer, the code is structured 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 40,000 words of storage and an LCM (or simulated LCM) of a few hundred thousand words is sufficient to eliminate the need for using random-access storage.

6. Running Time:

   Running time is directly related to problem size and to the computer's central processor and data transfer speeds.

On the Cray-XMP, a 69 energy group [with 40 thermal (upscatter) groups], S_{16}, P_3 scatter, 500 space-point k_{eff} calculation for a light water reactor requires about 48 sec CPU time.

A 30 energy group, S_{16}, P_4 scatter, 30 space-point, fixed surface source, detector efficiency problem requires about 1.4 sec CPU time on the Cray-XMP.

A 42 energy group, S_{32}, P_3 scatter, 200 space-point; coupled neutron/gamma shielding problem requires about 5 sec CPU time on the Cray-XMP.

Generally then, on the Cray-XMP, the running times for ONEDANT will range from a second to about a minute.

7. Unusual Features of the Program:

   The ONEDANT code 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. Thus, the package consists of an Input Module, an Edit Module, and the Solver Module. 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 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.

8. Programming Languages:

   The program is written in standard FORTRAN 77 language with only two known exceptions. There are a few variable names longer than six characters and in line comments are used. There are also a few C language routines used to interface to the Unix system.

9. Machine Requirements:

   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 200,000 card-images), sample problems, and the DANTSYS document, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System", LA-12969-M, have been submitted to the Radiation Shielding Information Center.

CODE ABSTRACT FOR TWODANT: A CODE PACKAGE FOR TWO-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT

by
Ray E. Alcouffe, Randal S. Baker, Forrest W. Brinkley,
Duane R. Marr, and R. Douglas O'Dell

TWODANT ABSTRACT

1. Program identification: TWODANT

2. Computer for which Program is designed:

   The current release is designed for Unix-like systems. The specific computers supported fall into two categories: Long word computers and short word computers. The program has been implemented on the long word Cray-YMP and Cray 2 computers. It has also been implemented on the Sun, SGI, HP9000, and IBM RS6000 short word workstations. The workstation versions use double precision arithmetic. Older versions have run on CDC Cyber 205, Vax, and main frame IBM but are not supported in this release.

3. Function:

   TWODANT solves the two-dimensional multigroup transport equation in x-y, r-z, and r-θ geometries. Both regular and adjoint, inhomogeneous (fixed source) and homogeneous (k_{eff} and eigenvalue search) problems subject to vacuum, reflective, periodic, white, or inhomogeneous boundary flux conditions are solved. General anisotropic scattering is allowed and anisotropic inhomogeneous sources are permitted.

4. Method of Solution:

   TWODANT numerically solves the two-dimensional, multigroup form of the neutral-particle, steady-state Boltzmann transport equation. The discrete-ordinates form of approximation is used for treating the angular variation of the particle distribution and the diamond-difference and adaptive weight diamond (AWDD) schemes are used for space-angle discretization. Negative fluxes are eliminated by a local set-to-zero-and-correct algorithm in the diamond case. 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. The diffusion solver uses the multigrid method and Chebychev acceleration of the fission source.

   A coupled Monte Carlo/discrete ordinates option for solving all or part of the problem using Monte Carlo is provided for x-y and r-z geometries. A ray trace first collision option to obtain a first collision source from an arbitrary source distribution may be used in x-y and r-z geometries.

5. Restrictions:

   The code is thoroughly variably dimensioned with a flexible, sophisticated data management and transfer capability. Originally designed for the CDC-7600 computer, the code is structured 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

40,000 words of storage and an LCM (or simulated LCM) of a few hundred thousand words is 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 Cray-XMP, a four group, adjoint calculation of the eigenvalue of an R-Z model of the Fast Test Reactor (FTR) took 15 seconds. The calculation used transport corrected P_0 cross sections, an S4 angular quadrature, and a 31 by 68 spatial mesh.

7. Unusual Features of the Program:

   The TWODANT code is modularly structured in a form that separates the input and the output (edit) functions from the main calculational (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 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.

8. Programming Languages:

   The program is written in standard FORTRAN 77 language with only two known exceptions. There are a few variable names longer than six characters and in line comments are used. There are also a few C language routines used to interface to the Unix system.

9. Machine Requirements:

   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 200,000 card-images), sample problems, and the DANTSYS document, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System", LA-12969-M, have been submitted to the Radiation Shielding Information Center.

CODE ABSTRACT FOR TWODANT/GQ: A CODE PACKAGE FOR TWO-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT ON A GENERALIZED QUADRILATERAL MESH

by
Ray E. Alcouffe, Forrest W. Brinkley, and Duane R. Marr

TWODANT/GQ ABSTRACT

1. Program identification: TWODANT/GQ

2. Computer for which Program is designed:
   Cray computers running UNICOS

3. Function:
   TWODANT/GQ solves the two-dimensional multigroup transport equation in a mesh of generalized quadrilaterals in x-y and r-z geometries. Both regular and adjoint, inhomogeneous (fixed source) and homogeneous (k_{eff} and eigenvalue search) problems subject to vacuum, reflective, periodic, white, or inhomogeneous boundary flux conditions are solved. General anisotropic scattering is allowed and anisotropic inhomogeneous sources are permitted.

4. Method of Solution:
   TWODANT/GQ numerically solves the two-dimensional, multigroup form of the neutral-particle, steady-state Boltzmann transport equation. The discrete-ordinates form of approximation is used for treating the angular variation of the particle distribution and the diamond-difference scheme is used for space-angle 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. The diffusion solver uses the multigrid method and Chebychev acceleration of the fission source.

5. Restrictions:
   The code is thoroughly variably dimensioned with a flexible, sophisticated data management and transfer capability. Originally designed for the CDC-7600 computer, the code is structured 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 40,000 words of storage and an LCM (or simulated LCM) of a few hundred thousand words is 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 Cray-YMP, a one group calculation of the eigenvalue of an x-y model of the sample TWODANT/GQ problem took 4.7 seconds. The calculation used transport corrected P_0 cross sections, an S4 square angular quadrature, and a 56 by 56 spatial mesh.

7. Unusual Features of the Program:

   The use of a generalized quadrilateral mesh allows accurate modeling of more complex shapes than can be accommodated by the usual rectilinear mesh. The TWODANT/GQ code is modularly structured in a form that separates the input and the output (edit) functions from the main calculational (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 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.

8. Programming Languages:

   The program is written in standard FORTRAN 77 language with only two known exceptions. There are a few variable names longer than six characters and in line comments are used. There are also a few C language routines used to interface to the Unix system.

9. Machine Requirements:

   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 200,000 card-images), sample problems, and the DANTSYS document, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System", LA-12969-M, have been submitted to the Radiation Shielding Information Center.

CODE ABSTRACT FOR TWOHEX: A CODE PACKAGE FOR TWO-DIMENSIONAL, NEUTRAL-PARTICLE TRANSPORT IN EQUILATERAL TRIANGULAR MESHES

by
Wallace F. Walters, Forrest W. Brinkley,
and Duane R. Marr

TWOHEX ABSTRACT

1. Program identification: TWOHEX

2. Computer for which Program is designed:
   The current release is designed for Unix-like systems. The specific computers supported fall into two categories: Long word computers and short word computers. The program has been implemented on the long word Cray-YMP and Cray 2 computers. It has also been implemented on the Sun, SGI, HP9000, and IBM RS6000 short word workstations. The workstation versions use double precision arithmetic. Older versions have run on CDC Cyber 205, Vax, and main frame IBM but are not supported in this release.

3. Function:
   TWOHEX solves the two-dimensional multigroup transport equation on an equilateral triangular mesh in the x,y plane. Both regular and adjoint, inhomogeneous (fixed source) and homogeneous problems are solved. Three problem domains are treated by TWOHEX. The "whole core" domain is a 60 degree parallelogram with vacuum boundary conditions on each face. The "third core" domain is a 120 degree parallelogram with two vacuum and two rotational boundary conditions. The "sixth core" domain is a 60 degree parallelogram with two vacuum and two rotational boundary conditions. General anisotropic scattering is allowed and an anisotropic inhomogeneous source may be input as cell averages.

4. Method of Solution:
   TWOHEX numerically solves the two-dimensional, multigroup form of the neutral-particle, steady-state Boltzmann transport equation. The discrete-ordinates form of approximation is used for treating the angular variation of the particle distribution and a linear characteristic/nodal scheme is used for spatial discretization. There is no negative flux fixup since few if any negative fluxes are generated by this numerical scheme. A standard inner (within-group) iteration, outer (energy-group-dependent source) iteration technique is used. Both inner and outer iterations are accelerated using the Chebyshev acceleration method.

5. Restrictions:
   The code is thoroughly variably dimensioned with a flexible, sophisticated data management and transfer capability. Originally designed for the CDC-7600 computer, the code is structured 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 40,000 words of storage and an LCM (or simulated LCM) of a few hundred thousand words is 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 a Cray-1S, a four group calculation of the eigenvalue of a midplane whole core model of the Fast Test Reactor (FTR) took 45 seconds. The calculation used transport corrected P0 cross sections, an S4 angular quadrature with 12 angles per hemisphere, and a 60 by 30 spatial mesh (six triangles per subassembly).

7. Unusual Features of the Program:

The triangular mesh allows accurate modeling of core geometries consisting of hexagonal fuel elements. The TWOHEX code is modularly structured in a form that separates the input and the output (edit) functions from the main calculational (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 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.

8. Programming Languages:

The program is written in standard FORTRAN 77 language with only two known exceptions. There are a few variable names longer than six characters and in line comments are used. There are also a few C language routines used to interface to the Unix system.

9. Machine Requirements:

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 200,000 card-images), sample problems, and the DANTSYS document, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System", LA-12969-M, have been submitted to the Radiation Shielding Information Center.

CODE ABSTRACT FOR THREEDANT: A CODE PACKAGE FOR THREE-DIMENSIONAL, DIFFUSION-ACCELERATED, NEUTRAL-PARTICLE TRANSPORT

by
Ray E. Alcouffe, Randal S. Baker, Forrest W. Brinkley, and Duane R. Marr

THREEDANT ABSTRACT

1. Program identification: THREEDANT

2. Computer for which Program is designed:

   The current release is designed for Unix-like systems. The specific computers supported fall into two categories: Long word computers and short word computers. The program has been implemented on the long word Cray-YMP and Cray 2 computers. It has also been implemented on the Sun, SGI, HP9000, and IBM RS6000 short word workstations. The workstation versions use double precision arithmetic. Older versions have run on CDC Cyber 205, Vax, and main frame IBM but are not supported in this release.

3. Function:

   THREEDANT solves the three-dimensional multigroup transport equation in x-y-z, and r-z-θ geometries. Both regular and adjoint, inhomogeneous (fixed source) and homogeneous (k_{eff} and eigenvalue search) problems subject to vacuum, reflective, periodic, white, or inhomogeneous boundary flux conditions are solved. General anisotropic scattering is allowed and anisotropic inhomogeneous sources are permitted.

4. Method of Solution:

   THREEDANT numerically solves the three-dimensional, multigroup form of the neutral-particle, steady-state Boltzmann transport equation. The discrete-ordinates form of approximation is used for treating the angular variation of the particle distribution and the diamond-difference scheme and adaptive weighted diamond (AWDD) schemes are used for space-angle discretization. Negative fluxes are eliminated by a local set-to-zero-and-correct algorithm in the diamond case. 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. The diffusion solver uses the multigrid method and Chebychev acceleration of the fission source.

5. Restrictions:

   The code is thoroughly variably dimensioned with a flexible, sophisticated data management and transfer capability. Originally designed for the CDC-7600 computer, the code is structured 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 40,000 words of storage and an LCM (or simulated LCM) of a few hundred thousand words is 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 Cray-YMP, a four group, adjoint calculation of the eigenvalue of an XYZ model of the Fast Test Reactor (FTR) took 35 seconds. The calculation used transport corrected P_0 cross sections, an S_8 angular quadrature, and a 14x14x30 spatial mesh.

7. Unusual Features of the Program:

   The THREEDANT code is modularly structured in a form that separates the input and the output (edit) functions from the main calculational (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 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.

8. Programming Languages:

   The program is written in standard FORTRAN 77 language with only two known exceptions. There are a few variable names longer than six characters and in line comments are used. There are also a few C language routines used to interface to the Unix system.

9. Machine Requirements:

   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 200,000 card-images), sample problems, and the DANTSYS document, "DANTSYS: A Diffusion Accelerated Neutral Particle Transport Code System", LA-12969-M, have been submitted to the Radiation Shielding Information Center.

BIBLIOGRAPHY

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

17

XTM — Transport Methods Group

Los Alamos
National Laboratory

BIBLIOGRAPHY

1. T. E. Albert and P. Nelson, "Computation of Azimuthally Dependent Albedo Data by Invariant Embedding," in Proc. of Sixth Intl. Conf. on Radiation Shielding, May 16-20, 1983, Tokyo, Vol. I, pp. 283-293.
2. R. E. Alcouffe, "A Diffusion Accelerated S_N Transport Method for Radiation Transport on a General Quadrilateral Mesh", *Nucl. Sci Eng*, **105**, 191-197,1990.
3. R. E. Alcouffe, R. D. O'Dell, and F. W. Brinkley, Jr., "A First-Collision Source Method That Satisfies Discrete S_n Transport Balance," *Nucl. Sci. Eng.* **105**, 198 (1990).
4. R. E. Alcouffe, "An Adaptive Weighted Diamond Differencing Method for Three-Dimensional XYZ Geometry," *Trans. Am. Nuc. Soc.* **68**, Part A, 206 (1993).
5. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," *Nucl. Sci. Eng.* **64**, 344 (1977).
6. R. E. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Multigroup Diffusion Equation," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 340-351.
7. R. E. Alcouffe, F. W. Brinkley, D. R. Marr, and R. D. O'Dell, "User's Guide for TWODANT: A Code Package for Two-Dimension, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory manual LA-10049-M, Rev. 1, (October 1984).
8. American National Standard Programming Language FORTRAN, ANSI X3.9-1978, American National Standards Institute, Inc., New York, NY 10018.
9. R. S. Baker, "A Fully Coupled Monte Carlo/Discrete Ordinates Solution to the Neutron Transport Equation," Ph.D. Dissertation, University of Arizona, Tucson, AZ (1990).
10. R. S. Baker, W. F. Filippone, and R. E. Alcouffe, "Extension of the Fully Coupled Monte Carlo/S_N Response Matrix Method to Problems Including Upscatter and Fission," Int. Topical Meeting on Advances in Math., Comp., and Reactor Phy., 5, p. 21.2 3-1, Pittsburgh, PA (1991).
11. R. S. Baker, W. F. Filippone, and R. E. Alcouffe, "The Multigroup and Radial Geometry Formulation of the Monte Carlo/S_N Response Matrix Method," *Nucl. Sci. Eng.* **105**, 184 (1990).
12. 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.
13. 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.
14. B. M. Carmichael, "Standard Interface Files and Procedures for Reactor Physics Codes, Version III," Los Alamos Scientific Laboratory report LA-5486-MS (February 1974).

15. K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," Argonne National Laboratory report ANL-82-64 (April 1984).

16. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693, (March 1967).

17. D. R. Ferguson and K. L. Derstine, "Optimized Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes," Nucl. Sci. Eng. **64**, 593 (1977).

18. W. F. Filippone, and R. E. Alcouffe, "The S_N/Monte Carlo Response Matrix Hybrid Method," *Nucl. Sci. Eng.* **100**, p. 209 (1988).

19. 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).

20. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).

21. R. D. O'Dell, F. W. Brinkley Jr., D. R. Marr, R. E. Alcouffe, "Revised User's Manual for ONEDANT: A Code Package for One-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory manual LA-9184-M, Rev. (December 1989).

22. W. A. Rhoades and R. L. Childs, "An Updated Version of the DOT4 One- and Two-Dimensional Neutron/Photon Transport Code," Oak Ridge National Laboratory report ORNL-5851, (July 1982).

23. W. A. Rhoades and F. R. Mynatt, "THE DOT III TWO-DIMENSIONAL DISCRETE ORDINATES TRANSPORT CODE," Oak Ridge National Laboratory report ORNL-TM-4280, (September 1973).

24. W. F. Walters, "The TLC Scheme for Numerical Solution of the Transport Equation on Equilateral Triangular Meshes," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 151-165.

25. W. F. Walters, F. W. Brinkley, and D. R. Marr, "User's Guide for TWOHEX: A Code Package for Two-Dimensional, Neutral-Particle Transport in Equilateral Triangular Meshes," Los Alamos National Laboratory manual LA-10258-M, (October 1984).

INDEX

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory

18

XTM — Transport Methods Group

Los Alamos
National Laboratory

A

B

C

D

E

F

G

H

I

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545