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)

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither The Regents of the University of California, the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by The Regents of the University of California, the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of The Regents of the University of California, the United States Government, or any agency thereof.

Los Alamos National Laboratory
An Affirmative Action/Equal Opportunity Employer

Los Alamos, New Mexico 87545
### Table of Contents

#### OneDant, Twodant, TwoHex, Twodant/GQ, and ThreeDant — Introduction and System Overview

- Table of Contents: 1-3
- List of Figures: 1-5
- Introduction: 1-6
- Documentation Summary: 1-8
- Common Modeling Concepts:
  - Geometry Concepts: 1-10
  - Iteration Strategy: 1-11
  - Modular Structure and Interface Files: 1-13
- References: 1-15

#### OneDant User’s Guide

- Table of Contents: 2-5
- List of Figures: 2-7
- List of Tables: 2-9
- Introduction: 2-11
- Documentation for OneDant Usage:
  - What Is In This User’s Guide: 2-13
  - What Is Available Elsewhere: 2-14
- OneDant Input Overview:
  - Input Block Order: 2-17
  - Free Field Input Summary: 2-19
  - MINI-MANUAL Introduction: 2-22
  - MINI MANUAL: 2-23
- OneDant Input Details:
  - Introduction: 2-27
  - Title Line Details: 2-30
  - Block-I Details: Dimensions and Controls: 2-31
  - Block-II Details: Geometry: 2-34
  - Block-III Details: Nuclear Data: 2-35
  - Block-IV Details: Cross-Section Mixing: 2-41
  - Block-V Details: Solver Input: 2-48
  - Block-VI Details: Edit Input: 2-58

Version 3.0
# TABLE OF CONTENTS

REFERENCES ..................................................................................................... 2-67

APPENDIX A: SAMPLE INPUT ...................................................................... 2-69
  Sample Problem 1: Standard $k_{eff}$ Calculation ..................................... 2-69
  Sample Problem 2: Edit-Only Run ............................................................ 2-86

APPENDIX B: OPERATING SYSTEM SPECIFICS ........................................ 2-93
  UNIX/UNICOS Execution ........................................................................ 2-93
  Library Search Path .................................................................................. 2-94

**TWODANT USER'S GUIDE** ........................................................................ 3-1

TABLE OF CONTENTS .................................................................................. 3-5
LIST OF FIGURES .......................................................................................... 3-7
LIST OF TABLES ............................................................................................ 3-9
INTRODUCTION ............................................................................................. 3-11

DOCUMENTATION FOR TWODANT USAGE ........................................... 3-13
  What Is In This User's Guide .................................................................... 3-13
  What Is Available Elsewhere ..................................................................... 3-14

TWODANT INPUT OVERVIEW ..................................................................... 3-17
  Input Block Order .................................................................................... 3-17
  Free Field Input Summary ....................................................................... 3-19
  MINI-MANUAL Introduction .................................................................... 3-22
  MINI MANUAL .......................................................................................... 3-23

TWODANT INPUT DETAILS ......................................................................... 3-29
  Introduction .............................................................................................. 3-29
  Title Line Details ..................................................................................... 3-32
  Block-I Details: Dimensions and Controls ............................................. 3-33
  Block-II Details: Geometry ...................................................................... 3-36
  Block-III Details: Nuclear Data ................................................................. 3-37
  Block-IV Details: Cross-Section Mixing ................................................ 3-43
  Block-V Details: Solver Input ................................................................... 3-50
  Block-VI Details: Edit Input ....................................................................... 3-65

REFERENCES ............................................................................................... 3-75

APPENDIX A: SAMPLE INPUT ..................................................................... 3-77
  Sample Problem 1: Standard $k_{eff}$ Calculation ..................................... 3-77
  Sample Problem 1: Output Description .................................................... 3-77
  Sample Problem 1: Output Listing ............................................................ 3-79
  Sample Problem 2: Coupled $S_n$ Monte Carlo Calculation .................... 3-97
  Sample Problem 2: Output Description .................................................... 3-97
  Sample Problem 2: Output Listing ............................................................ 3-102

APPENDIX B: OPERATING SYSTEM SPECIFICS ....................................... 3-123
  UNIX/UNICOS Execution ........................................................................ 3-123
  Library Search Path .................................................................................. 3-124
# TABLE OF CONTENTS

## TWODANT/GQ USER'S GUIDE

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>4-5</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>4-7</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>4-9</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>4-11</td>
</tr>
<tr>
<td>DOCUMENTATION FOR TWODANT/GQ USAGE</td>
<td>4-13</td>
</tr>
<tr>
<td>What Is In This User's Guide</td>
<td>4-13</td>
</tr>
<tr>
<td>What Is Available Elsewhere</td>
<td>4-14</td>
</tr>
<tr>
<td>GEOMETRY CONCEPTS</td>
<td>4-17</td>
</tr>
<tr>
<td>Submeshes</td>
<td>4-18</td>
</tr>
<tr>
<td>Objects</td>
<td>4-19</td>
</tr>
<tr>
<td>Components</td>
<td>4-20</td>
</tr>
<tr>
<td>Geometry</td>
<td>4-20</td>
</tr>
<tr>
<td>Boundary Conditions</td>
<td>4-21</td>
</tr>
<tr>
<td>TWODANT/GQ INPUT OVERVIEW</td>
<td>4-23</td>
</tr>
<tr>
<td>Input Block Order</td>
<td>4-23</td>
</tr>
<tr>
<td>Free Field Input Summary</td>
<td>4-25</td>
</tr>
<tr>
<td>MINI-MANUAL Introduction</td>
<td>4-28</td>
</tr>
<tr>
<td>MINI MANUAL</td>
<td>4-29</td>
</tr>
<tr>
<td>TWODANT/GQ INPUT DETAILS</td>
<td>4-33</td>
</tr>
<tr>
<td>Introduction</td>
<td>4-33</td>
</tr>
<tr>
<td>Title Line Details</td>
<td>4-36</td>
</tr>
<tr>
<td>Block-I Details: Dimensions and Controls</td>
<td>4-37</td>
</tr>
<tr>
<td>Block-II Details: Geometry</td>
<td>4-39</td>
</tr>
<tr>
<td>Block-III Details: Nuclear Data</td>
<td>4-46</td>
</tr>
<tr>
<td>Block-IV Details: Cross-Section Mixing</td>
<td>4-52</td>
</tr>
<tr>
<td>Block-V Details: Solver Input</td>
<td>4-59</td>
</tr>
<tr>
<td>Block-VI Details: Edit Input</td>
<td>4-68</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>4-79</td>
</tr>
<tr>
<td>APPENDIX A: SAMPLE INPUT</td>
<td>4-81</td>
</tr>
<tr>
<td>Sample Problem: Supercell k\textsubscript{eff} Calculation</td>
<td>4-81</td>
</tr>
<tr>
<td>Sample Problem: Output Description</td>
<td>4-82</td>
</tr>
<tr>
<td>Sample Problem: Output Listing</td>
<td>4-85</td>
</tr>
<tr>
<td>APPENDIX B: OPERATING SYSTEM SPECIFICS</td>
<td>4-95</td>
</tr>
<tr>
<td>UNIX/UNICOS Execution</td>
<td>4-95</td>
</tr>
<tr>
<td>Library Search Path</td>
<td>4-96</td>
</tr>
</tbody>
</table>

## TWOHEX USER'S GUIDE

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>5-5</td>
</tr>
</tbody>
</table>
**TABLE OF CONTENTS**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>5-7</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>5-9</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>5-11</td>
</tr>
<tr>
<td>DOCUMENTATION FOR TWOHEX USAGE</td>
<td>5-13</td>
</tr>
<tr>
<td>What Is In This User’s Guide</td>
<td>5-13</td>
</tr>
<tr>
<td>What Is Available Elsewhere</td>
<td>5-14</td>
</tr>
<tr>
<td>TWOHEX INPUT OVERVIEW</td>
<td>5-17</td>
</tr>
<tr>
<td>Input Block Order</td>
<td>5-17</td>
</tr>
<tr>
<td>Free Field Input Summary</td>
<td>5-19</td>
</tr>
<tr>
<td>MINI-MANUAL Introduction</td>
<td>5-22</td>
</tr>
<tr>
<td>MINI MANUAL</td>
<td>5-23</td>
</tr>
<tr>
<td>TWOHEX INPUT DETAILS</td>
<td>5-27</td>
</tr>
<tr>
<td>Introduction</td>
<td>5-27</td>
</tr>
<tr>
<td>Title Line Details</td>
<td>5-30</td>
</tr>
<tr>
<td>Block-I Details: Dimensions and Controls</td>
<td>5-31</td>
</tr>
<tr>
<td>Block-II Details: Geometry</td>
<td>5-33</td>
</tr>
<tr>
<td>Block-III Details: Nuclear Data</td>
<td>5-34</td>
</tr>
<tr>
<td>Block-IV Details: Cross-Section Mixing</td>
<td>5-40</td>
</tr>
<tr>
<td>Block-V Details: Solver Input</td>
<td>5-45</td>
</tr>
<tr>
<td>Block-VI Details: Edit Input</td>
<td>5-50</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>5-59</td>
</tr>
<tr>
<td>APPENDIX A: SAMPLE INPUT</td>
<td>5-61</td>
</tr>
<tr>
<td>Sample Problem: Standard k\textsubscript{eff} Calculation</td>
<td>5-61</td>
</tr>
<tr>
<td>Sample Problem: Output Description</td>
<td>5-62</td>
</tr>
<tr>
<td>APPENDIX B: OPERATING SYSTEM SPECIFICS</td>
<td>5-75</td>
</tr>
<tr>
<td>UNIX/UNICOS Execution</td>
<td>5-75</td>
</tr>
<tr>
<td>Library Search Path</td>
<td>5-76</td>
</tr>
</tbody>
</table>

**THREEDANT USER’S GUIDE** ........................................................................ 6-1

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>6-5</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>6-7</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>6-9</td>
</tr>
<tr>
<td>DOCUMENTATION FOR THREEDANT USAGE</td>
<td>6-13</td>
</tr>
<tr>
<td>What Is In This User’s Guide</td>
<td>6-13</td>
</tr>
<tr>
<td>What Is Available Elsewhere</td>
<td>6-14</td>
</tr>
<tr>
<td>THREEDANT INPUT OVERVIEW</td>
<td>6-17</td>
</tr>
<tr>
<td>Input Block Order</td>
<td>6-17</td>
</tr>
<tr>
<td>Free Field Input Summary</td>
<td>6-19</td>
</tr>
<tr>
<td>MINI-MANUAL Introduction</td>
<td>6-22</td>
</tr>
</tbody>
</table>
**TABLE OF CONTENTS**

MINI MANUAL ............................................................................................. 6-23

**THREEDANT INPUT DETAILS** ................................................................. 6-29
Introduction .................................................................................................. 6-29
Title Line Details ......................................................................................... 6-32
Block-I Details: Dimensions and Controls ................................................. 6-33
Block-II Details: Geometry ......................................................................... 6-36
Block-III Details: Nuclear Data .................................................................... 6-37
Block-IV Details: Cross-Section Mixing ..................................................... 6-43
Block-V Details: Solver Input ...................................................................... 6-50
Block-VI Details: Edit Input ....................................................................... 6-61

REFERENCES ............................................................................................... 6-73

APPENDIX A: SAMPLE INPUT .................................................................... 6-75
Sample Problem: Standard $k_{eff}$ Calculation ........................................... 6-75
Sample Problem: Output Description .......................................................... 6-75
Sample Problem: Output Listing ................................................................. 6-77

APPENDIX B: OPERATING SYSTEM SPECIFICS ...................................... 6-97
UNIX/UNICOS Execution ........................................................................... 6-97
Library Search Path ..................................................................................... 6-98

**DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT** .......... 7-1

**TABLE OF CONTENTS** ......................................................................... 7-3

LIST OF FIGURES ......................................................................................... 7-5

LIST OF TABLES ........................................................................................... 7-7

INTRODUCTION .......................................................................................... 7-9

MORE DETAILS ON BLOCK-I INPUT .......................................................... 7-11
Angular Quadrature (ISN) ........................................................................... 7-11
Geometry (NZONE, IM, IT) ......................................................................... 7-11
MAXSCM, MAXLCM .................................................................................. 7-11
Execution/File Suppression Flags ............................................................... 7-12

MORE DETAILS ON GEOMETRY INPUT ...................................................... 7-13

MORE DETAILS ON SOLVER INPUT .......................................................... 7-15
Iteration Strategy ......................................................................................... 7-15
Convergence Criteria ................................................................................... 7-19
Grey Acceleration of Upscatter ................................................................. 7-21
Iteration Monitor Print ................................................................................. 7-21
Boundary Conditions ................................................................................... 7-23
Input of Quadrature Sets .............................................................................. 7-24
Zone-Dependent Fission Fractions (the CHI Array) ..................................... 7-25
Input of Inhomogeneous Sources ............................................................... 7-26
Normalization of the Calculation (the NORM Parameter) ......................... 7-30
Transport Corrections for the Cross Sections (TRCOR) ............................. 7-31
Buckling Corrections .................................................................................... 7-33

Version 3.0
<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenvalue Searches</td>
<td>7-33</td>
</tr>
<tr>
<td>Adjoint Computations</td>
<td>7-37</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>7-39</td>
</tr>
</tbody>
</table>

**RUNNING THE EDIT MODULE** .......................................................... 8-1

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>8-3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>8-5</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>8-7</td>
</tr>
<tr>
<td>REACTION RATES</td>
<td>8-9</td>
</tr>
<tr>
<td>Spatial Options for Edits</td>
<td>8-9</td>
</tr>
<tr>
<td>Energy Group Options For Edits</td>
<td>8-10</td>
</tr>
<tr>
<td>Forms Of Response Functions</td>
<td>8-11</td>
</tr>
<tr>
<td>Response Function Summing Options</td>
<td>8-13</td>
</tr>
<tr>
<td>Adjoint Edits</td>
<td>8-15</td>
</tr>
<tr>
<td>ASCII File Output Capabilities (the EDOUTF Parameter)</td>
<td>8-15</td>
</tr>
<tr>
<td>MASS INVENTORIES</td>
<td>8-17</td>
</tr>
</tbody>
</table>

**FREE FIELD INPUT REFERENCE** ........................................................... 9-1

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>9-3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>9-5</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>9-7</td>
</tr>
<tr>
<td>FREE FIELD DEFINITIONS</td>
<td>9-9</td>
</tr>
<tr>
<td>Arrays</td>
<td>9-9</td>
</tr>
<tr>
<td>Numeric Data Items</td>
<td>9-9</td>
</tr>
<tr>
<td>Character Data Items</td>
<td>9-10</td>
</tr>
<tr>
<td>Blocks</td>
<td>9-10</td>
</tr>
<tr>
<td>Strings</td>
<td>9-10</td>
</tr>
<tr>
<td>Comments</td>
<td>9-10</td>
</tr>
<tr>
<td>Operators</td>
<td>9-10</td>
</tr>
<tr>
<td>FREE FIELD INPUT DETAILS</td>
<td>9-13</td>
</tr>
<tr>
<td>Free-Field Input</td>
<td>9-13</td>
</tr>
<tr>
<td>User-Specified Input Formats</td>
<td>9-16</td>
</tr>
<tr>
<td>FIXED-FIELD FIDO INPUT DETAILS</td>
<td>9-19</td>
</tr>
<tr>
<td>Fixed-Field FIDO Input</td>
<td>9-19</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>9-23</td>
</tr>
</tbody>
</table>

**CROSS-SECTION LIBRARIES** ............................................................. 10-1

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>10-3</td>
</tr>
</tbody>
</table>

---

Version 3.0
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>10-5</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>10-7</td>
</tr>
<tr>
<td>INPUT OF THE BASIC CROSS-SECTION LIBRARY</td>
<td>10-9</td>
</tr>
<tr>
<td>ISOTXS and GRUPXS Standard Interface Files</td>
<td>10-9</td>
</tr>
<tr>
<td>Card-Image Libraries in Los Alamos, ANISN, or Free Field Format</td>
<td>10-9</td>
</tr>
<tr>
<td>Binary Form of Card-Image Libraries (the BXSLIB file)</td>
<td>10-12</td>
</tr>
<tr>
<td>XSLIBB Card-Image Library File</td>
<td>10-12</td>
</tr>
<tr>
<td>MACRXS and SNXEDT Cross-Section Files</td>
<td>10-13</td>
</tr>
<tr>
<td>The MACBCD Card-Image Cross-Section Library</td>
<td>10-13</td>
</tr>
<tr>
<td>The Los Alamos MENDF5 Cross-Section Library</td>
<td>10-13</td>
</tr>
<tr>
<td>The Los Alamos MENDF5G Gamma Cross-Section Library</td>
<td>10-14</td>
</tr>
<tr>
<td>The XSLIBE and XSLIBF Material Cross-Section Libraries</td>
<td>10-14</td>
</tr>
<tr>
<td>COUPLED NEUTRON-GAMMA CROSS SECTIONS</td>
<td>10-15</td>
</tr>
<tr>
<td>CREATING FILES WITH DIFFERENT FORMATS</td>
<td>10-19</td>
</tr>
<tr>
<td>BALANCING THE CROSS SECTIONS</td>
<td>10-21</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>10-23</td>
</tr>
</tbody>
</table>

MATERIAL MIXING TUTORIAL .................................................................................. 11-1

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>11-3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>11-5</td>
</tr>
<tr>
<td>REVIEW OF TERMINOLOGY</td>
<td>11-7</td>
</tr>
<tr>
<td>MIXING MATERIALS FROM &quot;ISOTOPES&quot;</td>
<td>11-9</td>
</tr>
<tr>
<td>ASSIGNING MATERIALS TO ZONES</td>
<td>11-11</td>
</tr>
<tr>
<td>ALTERNATIVE FORMS OF MIXING</td>
<td>11-13</td>
</tr>
<tr>
<td>Using Atomic Fractions or Weight Fractions (MATSPEC)</td>
<td>11-13</td>
</tr>
<tr>
<td>Providing Atomic Weights to the Code (the ATWT Array)</td>
<td>11-15</td>
</tr>
<tr>
<td>INTERFACE FILES USED IN MIXING</td>
<td>11-17</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>11-19</td>
</tr>
</tbody>
</table>

ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual .................................. 12-1

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>12-3</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>12-5</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>12-7</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>12-9</td>
</tr>
<tr>
<td>The First Order Form of the Boltzmann Transport Equation</td>
<td>12-9</td>
</tr>
<tr>
<td>MULTIGROUP, DISCRETE ORDINATES TRANSPORT THEORY</td>
<td>12-11</td>
</tr>
<tr>
<td>Multigroup Approximation</td>
<td>12-11</td>
</tr>
</tbody>
</table>

Version 3.0
# TABLE OF CONTENTS

Discrete Ordinates Approximation ............................................................... 12-12  
Iteration Procedure and Diffusion Synthetic Acceleration .............................. 12-14  
ONEDANT METHODS ..................................................................................... 12-19  
   Geometrical Symmetries Treated in ONEDANT ......................................... 12-19  
TWODANT METHODS .................................................................................... 12-35  
   Some Angular Details in TWODANT .................................................. ........ 12-35  
   Transport Operator in Two-Dimensional Symmetries ................................. 12-36  
   Spatially Discretized Two-Dimensional Transport Equation ....................... 12-37  
   Monte Carlo/Discrete Ordinates Hybrid Method ......................................... 12-40  
TWODANT/GQ METHODS ............................................................................. 12-49  
TWOHEX METHODS ....................................................................................... 12-51  
THREEDANT METHODS ................................................................................ 12-53  
REFERENCES ................................................................................................... 12-55  

ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT —  
Code Structure .................................................................................................. 13-1  
TABLE OF CONTENTS...................................................................................... 13-3  
LIST OF TABLES ................................................................................................ 13-5  
OPERATION OF THE CODE SYSTEM ............................................................ 13-7  
   Programming Practices and Standards........................................................ 13-7  
CODE PACKAGE STRUCTURE ....................................................................... 13-9  
   Input Module ............................................................................................... 13-16  
   Solver Modules ............................................................................................ 13-17  
   Edit Module .................................................................................................. 13-17  
PIECEWISE EXECUTION ................................................................................ 13-19  
   Module Execution Control ............................................................................ 13-19  
   Submodule Execution Control (File Generation Suppression) ..................... 13-20  
STACKED RUNS .............................................................................................. 13-25  
REFERENCES ................................................................................................... 13-27  

ERROR MESSAGES .......................................................................................... 14-1  
TABLE OF CONTENTS...................................................................................... 14-3  
INPUT ERRORS ................................................................................................ 14-5  
   Examples of Errors and Resulting Messages ................................................ 14-5  
COMMENTS REGARDING MULTIPLE ERRORS ........................................ 14-11  
IMPLEMENTATION ERRORS ......................................................................... 14-13  
WARNINGS ....................................................................................................... 14-15
FILE DESCRIPTIONS .......................................................................................... 15-1

TABLE OF CONTENTS ...................................................................................... 15-3

INTRODUCTION ............................................................................................... 15-5

ASCII FILES .................................................................................................... 15-7
Problem Input File .......................................................................................... 15-7
Cross-Section Library Files ............................................................................ 15-7
EDTOUT ......................................................................................................... 15-13
EDTOGX ...................................................................................................... 15-17
ARBFLUX .................................................................................................... 15-17

STANDARD INTERFACE FILES .................................................................... 15-19

CODE DEPENDENT INTERFACE FILES ...................................................... 15-21
AAFLXM for TWODANT ........................................................................... 15-22
AAFLXM for THREEDANT ................................................................. 15-24
ADJMPC ...................................................................................................... 15-28
AMFLUX .................................................................................................... 15-31
ASGMAT ...................................................................................................... 15-33
AZMFLX .................................................................................................... 15-36
BXSLIB ........................................................................................................ 15-38
EDIIT .......................................................................................................... 15-40
FISSRC ......................................................................................................... 15-47
GEODST ...................................................................................................... 15-49
GEOSING ..................................................................................................... 15-61
LNK3DNT .................................................................................................... 15-62
MACRXS ...................................................................................................... 15-64
RAFLXM for TWODANT ........................................................................... 15-67
RAFLXM for THREEDANT ................................................................. 15-70
RMFLUX .................................................................................................... 15-74
RZMFLX .................................................................................................... 15-76
SNXEDT ....................................................................................................... 15-78
SOLINP ........................................................................................................ 15-81
UCFLUX ....................................................................................................... 15-96

OVERWRITTEN INPUT FILES ....................................................................... 15-99
REFERENCES ................................................................................................. 15-101

CODE ABSTRACTS ....................................................................................... 16-1

INTRODUCTION .............................................................................................. 16-3
ONEDANT ABSTRACT .................................................................................. 16-7
TWODANT ABSTRACT .................................................................................. 16-11
TWODANT/GQ ABSTRACT ........................................................................... 16-15
TWOHEX ABSTRACT .................................................................................... 16-19
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Documentation Elements</td>
<td>1-8</td>
</tr>
<tr>
<td>2.1</td>
<td>LIBNAME Availability</td>
<td>2-37</td>
</tr>
<tr>
<td>2.2</td>
<td>UNIX Search Path</td>
<td>2-94</td>
</tr>
<tr>
<td>3.1</td>
<td>LIBNAME Availability</td>
<td>3-39</td>
</tr>
<tr>
<td>3.2</td>
<td>UNIX Search Path</td>
<td>3-124</td>
</tr>
<tr>
<td>4.1</td>
<td>LIBNAME Availability</td>
<td>4-48</td>
</tr>
<tr>
<td>4.2</td>
<td>UNIX Search Path</td>
<td>4-96</td>
</tr>
<tr>
<td>5.1</td>
<td>LIBNAME Availability</td>
<td>5-36</td>
</tr>
<tr>
<td>5.2</td>
<td>UNIX Search Path</td>
<td>5-76</td>
</tr>
<tr>
<td>6.1</td>
<td>LIBNAME Availability</td>
<td>6-39</td>
</tr>
<tr>
<td>6.2</td>
<td>UNIX Search Path</td>
<td>6-98</td>
</tr>
<tr>
<td>7.1</td>
<td>Source Ordering Index in Slab Geometry</td>
<td>7-29</td>
</tr>
<tr>
<td>7.2</td>
<td>Source Ordering Index in Cylindrical Geometry</td>
<td>7-30</td>
</tr>
<tr>
<td>8.1</td>
<td>MASSED Input Values</td>
<td>8-17</td>
</tr>
<tr>
<td>8.2</td>
<td>MASSED Input Values for Fine Mesh Mix Problem</td>
<td>8-18</td>
</tr>
<tr>
<td>9.1</td>
<td>Free Field Data Operators</td>
<td>9-15</td>
</tr>
<tr>
<td>9.2</td>
<td>Special Fixed Field Data Operators</td>
<td>9-21</td>
</tr>
<tr>
<td>10.1</td>
<td>Cross-Section Ordering In A Card-Image Library</td>
<td>10-10</td>
</tr>
<tr>
<td>10.2</td>
<td>Arrangement of Data in a Coupled Neutron-Gamma Library Table</td>
<td>10-16</td>
</tr>
<tr>
<td>10.3</td>
<td>Coupled Cross-Section Table Example</td>
<td>10-17</td>
</tr>
<tr>
<td>11.1</td>
<td>Example Specification of Materials</td>
<td>11-10</td>
</tr>
<tr>
<td>11.2</td>
<td>Composition of Zones</td>
<td>11-11</td>
</tr>
<tr>
<td>12.1</td>
<td>Forms of the Divergence Operator</td>
<td>12-21</td>
</tr>
<tr>
<td>12.2</td>
<td>Number of Spherical Harmonics, N, as a Function of Order</td>
<td>12-24</td>
</tr>
<tr>
<td>12.3</td>
<td>Spherical Harmonics, for Different Geometries</td>
<td>12-25</td>
</tr>
<tr>
<td>12.4</td>
<td>Number of Quadrature Points, M as a Function of S_n Order, N</td>
<td>12-27</td>
</tr>
<tr>
<td>12.5</td>
<td>Spherical Harmonics in Two Dimensions</td>
<td>12-35</td>
</tr>
<tr>
<td>12.6</td>
<td>Number of Angles Per Octant in Two Dimensions</td>
<td>12-36</td>
</tr>
<tr>
<td>13.1</td>
<td>Files Read and Written</td>
<td>13-9</td>
</tr>
<tr>
<td>13.2</td>
<td>Structure of the Input Module</td>
<td>13-12</td>
</tr>
<tr>
<td>13.3</td>
<td>Structure of the Onedant Solver Module</td>
<td>13-13</td>
</tr>
<tr>
<td>13.4</td>
<td>Structure of the Twodant Solver Module</td>
<td>13-13</td>
</tr>
<tr>
<td>13.5</td>
<td>Structure of the Threedant Solver Module</td>
<td>13-14</td>
</tr>
<tr>
<td>13.6</td>
<td>Structure of the Edit Module</td>
<td>13-15</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1: Spatial mesh labeling convention in DANTSYS ......................... 1-10
Figure 1.2: Simplified flow diagram of SOLVER iteration strategy .......... 1-12
Figure 1.3: DANTSYS Structure ......................................................... 1-13
Figure 2.1: ONEDANT Input Order .................................................... 2-18
Figure 3.1: TWODANT Input Order ................................................... 3-18
Figure 4.1: Submesh Examples .......................................................... 4-18
Figure 4.2: Examples of Objects ......................................................... 4-19
Figure 4.3: A Component Formed from Four Objects ......................... 4-20
Figure 4.4: The Final Geometry .......................................................... 4-21
Figure 4.5: Example of a Boundary Segment .................................... 4-22
Figure 4.6: TWODANT/GQ Input Order ............................................. 4-24
Figure 4.7: Hypothetical Supercell Geometry .................................... 4-81
Figure 5.1: TWOHEX Input Order ..................................................... 5-18
Figure 5.2: Possible $S_6$ Quadrature Arrangements ......................... 5-48
Figure 5.3: Core Map of the Sample Problem ................................... 5-61
Figure 5.4: Mesh Model for the Sample Problem ............................... 5-61
Figure 6.1: THREEDANT Input Order ............................................... 6-18
Figure 6.2: Orientation of Faces ........................................................ 6-59
Figure 7.1: Simplified flow diagram of SOLVER iteration strategy ....... 7-18
Figure 7.2: Ordering in slab geometry ............................................... 7-28
Figure 7.3: Quadrature points in cylindrical geometry ....................... 7-30
Figure 7.4: Variation of $\lambda$ during a hypothetical eigenvalue search ... 7-37
Figure 12.1: Coordinates in plane geometry ...................................... 12-19
Figure 12.2: Coordinates in cylindrical geometry ............................. 12-20
Figure 12.3: Coordinates in spherical geometry ............................... 12-20
Figure 12.4: Ordering of $S_6$ directions in plane and spherical geometries 12-28
Figure 12.5: Ordering of $S_4$ directions in two-angle plane geometry .... 12-29
Figure 12.6: Ordering of $S_6$ directions in cylindrical geometry .......... 12-30
ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT—Introduction and System Overview

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory
TABLE OF CONTENTS

TABLE OF CONTENTS ................................................................. 1-3
LIST OF FIGURES ................................................................. 1-5
INTRODUCTION ................................................................. 1-6
DOCUMENTATION SUMMARY ............................................... 1-8
COMMON MODELING CONCEPTS ........................................... 1-10
  Geometry Concepts ......................................................... 1-10
  Iteration Strategy ........................................................... 1-11
  Modular Structure and Interface Files ................................. 1-13
REFERENCES ................................................................. 1-15
LIST OF FIGURES

Figure 1.1: Spatial mesh labeling convention in DANTSYS. ................................. 1-10
Figure 1.2: Simplified flow diagram of SOLVER iteration strategy ....................... 1-12
Figure 1.3: DANTSYS Structure ........................................................................ 1-13
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;
INTRODUCTION

- 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\textsubscript{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\textsubscript{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.
The $S_n$ Code package includes documentation for varying audiences.

Table 1.1 Documentation Elements

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

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. Also included are the published manuals for TWODANT and TWOHEX.

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.](image)

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 $k_{\text{eff}}$. Also the alpha eigenvalue (time constant) of the system is determined by a search procedure based upon successive determinations of $k_{\text{eff}}$.

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.

![Diagram of DANTSYS Structure](image)

**Figure 1.3** DANTSYS Structure

More detail on this topic will be found in the chapter “ONEDANT, TWODANT, TWO-HEX, TWODANT/GQ, and THREEDANT — Code Structure”.

OVERVIEW  Version 3.0
REFERENCES


ONEDANT USER'S GUIDE

Deterministic Transport Team
Transport Methods Group, XTM
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

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

This work was supported by the US Department of Energy.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
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

TABLE OF CONTENTS........................................................................................................ 2-5

LIST OF FIGURES ............................................................................................................ 2-7

LIST OF TABLES .............................................................................................................. 2-9

INTRODUCTION .................................................................................................................. 2-11

DOCUMENTATION FOR ONEDANT USAGE .................................................................... 2-13
  What Is In This User's Guide .......................................................................................... 2-13
  What Is Available Elsewhere ......................................................................................... 2-14

ONEDANT INPUT OVERVIEW .......................................................................................... 2-17
  Input Block Order ......................................................................................................... 2-17
  Free Field Input Summary ............................................................................................ 2-19
  Arrays ............................................................................................................................ 2-19
  Numeric Data Items ...................................................................................................... 2-19
  Character Data Items .................................................................................................. 2-19
  Blocks .......................................................................................................................... 2-20
  Strings ......................................................................................................................... 2-20
  Comments ................................................................................................................... 2-20
  Operators ..................................................................................................................... 2-20
  FrequentlyUsed Operators ........................................................................................... 2-21

MINI-MANUAL Introduction ............................................................................................ 2-22

MINI-MANUAL .................................................................................................................... 2-23

ONEDANT INPUT DETAILS ............................................................................................... 2-27
  Introduction .................................................................................................................. 2-27
  Title Line Details ......................................................................................................... 2-30
    Title Line Control ..................................................................................................... 2-30
  Block-I Details: Dimensions and Controls ..................................................................... 2-31
    Dimensions ................................................................................................................ 2-31
    Storage Requirements ............................................................................................... 2-32
    Minimum Print Output .............................................................................................. 2-32
    Run Configuration Controls ..................................................................................... 2-33
  Block-II Details: Geometry ........................................................................................... 2-34
    Geometry Arrays ....................................................................................................... 2-34
  Block-III Details: Nuclear Data ....................................................................................... 2-35
    Nuclear Data Type and Options ................................................................................ 2-35
    Alternate Library Name ............................................................................................ 2-37
    Text Cross-Section Library Format ......................................................................... 2-39
  Block-IV Details: Cross-Section Mixing ....................................................................... 2-41
    MATLS input array .................................................................................................... 2-41
    Primary Mixing Arrays .............................................................................................. 2-42
    ASSIGN input array ................................................................................................. 2-43
    PREMIX input array ................................................................................................. 2-43
    Character Names vs. Numeric Names ........................................................................ 2-44
    Mixing Array for a Concentration Search .................................................................. 2-44
    ASGMOD input array ................................................................................................. 2-45
    Concentration Modifier ............................................................................................. 2-45
    Miscellaneous Mixing Input ....................................................................................... 2-47
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block-V Details: Solver Input</td>
<td>2-48</td>
</tr>
<tr>
<td>Desired Calculation</td>
<td>2-48</td>
</tr>
<tr>
<td>Iteration Controls</td>
<td>2-49</td>
</tr>
<tr>
<td>Acceleration Controls</td>
<td>2-49</td>
</tr>
<tr>
<td>K-Code Convergence</td>
<td>2-50</td>
</tr>
<tr>
<td>Output Controls</td>
<td>2-50</td>
</tr>
<tr>
<td>Miscellaneous Solver Input</td>
<td>2-51</td>
</tr>
<tr>
<td>Quadrature Details</td>
<td>2-52</td>
</tr>
<tr>
<td>Flux Guess From RTFLUX</td>
<td>2-52</td>
</tr>
<tr>
<td>General Eigenvalue Search Control</td>
<td>2-53</td>
</tr>
<tr>
<td>Dimension Search Input</td>
<td>2-53</td>
</tr>
<tr>
<td>Concentration Search Input</td>
<td>2-54</td>
</tr>
<tr>
<td>Volumetric Source Options</td>
<td>2-55</td>
</tr>
<tr>
<td>Boundary Source Input</td>
<td>2-56</td>
</tr>
<tr>
<td>Albedos</td>
<td>2-57</td>
</tr>
<tr>
<td>Block-VI Details: Edit Input</td>
<td>2-58</td>
</tr>
<tr>
<td>Spatial Specifications for Reaction Rates</td>
<td>2-58</td>
</tr>
<tr>
<td>Reaction Rates from Cross Sections</td>
<td>2-59</td>
</tr>
<tr>
<td>Edit Cross-Section Types by Position and Name</td>
<td>2-60</td>
</tr>
<tr>
<td>Reaction Rates from User Response Functions</td>
<td>2-61</td>
</tr>
<tr>
<td>Energy Group Collapse Specifications</td>
<td>2-62</td>
</tr>
<tr>
<td>Reaction Rate Summing</td>
<td>2-63</td>
</tr>
<tr>
<td>Mass Inventories</td>
<td>2-63</td>
</tr>
<tr>
<td>Power Normalization</td>
<td>2-64</td>
</tr>
<tr>
<td>Miscellaneous Edit Items</td>
<td>2-65</td>
</tr>
<tr>
<td>MENDF Library Edit Cross Sections</td>
<td>2-66</td>
</tr>
</tbody>
</table>

**REFERENCES** ........................................................................................................... 2-67

**APPENDIX A: SAMPLE INPUT** ................................................................. 2-69

Sample Problem 1: Standard $k_{eff}$ Calculation ........................................ 2-69
Sample Problem 2: Edit-Only Run ................................................................. 2-86

**APPENDIX B: OPERATING SYSTEM SPECIFICS** .............................................. 2-93

UNIX/UNICOS Execution ...................................................................................... 2-93
Library Search Path ............................................................................................. 2-94
LIST OF FIGURES

Figure 2.1: ONEDANT Input Order ................................................................. 2-18
LIST OF TABLES

Table 2.1: LIBNAME Availability ................................................................. 2-37
Table 2.2: UNIX Search Path ................................................................. 2-94
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.\textsuperscript{1,2}

ONEDANT\textsuperscript{TM} is based on the modular construction of the DANTSYS\textsuperscript{TM} 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\textsuperscript{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\textsuperscript{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 $k_{\text{eff}}$ calculation options, as well as time-absorption ($\alpha$), nuclide concentration, or dimensional search options,

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

\[ n \text{ O } d \]

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:

\[ \text{SOURCX= } 0 \text{ } 0 \text{ } 0 \text{ } 2R \text{ } 1.1 \text{ } F0 \]

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 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

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

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
(316 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
NOEDITT
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
I2LPI
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

<table>
<thead>
<tr>
<th>MATNAM [MT]</th>
<th>ZONNAM [NZONE]</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATSPEC [-]</td>
<td>valid: ATFRAC</td>
</tr>
<tr>
<td></td>
<td>WTRAC</td>
</tr>
<tr>
<td></td>
<td>ATDEN</td>
</tr>
</tbody>
</table>

| 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; -]
RSFNAME [-]

ICOLL [K], K ≤ NGROUP
IGRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

T
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}

<table>
<thead>
<tr>
<th>Word</th>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHEAD</td>
<td>Number of title lines that follow.(^{b})</td>
</tr>
</tbody>
</table>
| 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGEOm</td>
<td>Geometry. 1/2/3 = planar/cylindrical/spherical. or use one of the following character strings: PLANE, CYLINDER, CYL, SPHERE, SPH, SLAB, SLAB2ANG&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>NGROUP</td>
<td>Number of energy groups.</td>
</tr>
<tr>
<td>ISN</td>
<td>S&lt;sub&gt;n&lt;/sub&gt; order to be used. Must be even.</td>
</tr>
<tr>
<td>NISO</td>
<td>Number of isotopes on the basic input cross-section library.</td>
</tr>
<tr>
<td>MT</td>
<td>Number of materials&lt;sup&gt;b&lt;/sup&gt; in the problem (mixed from the isotopes).</td>
</tr>
<tr>
<td>NZONE</td>
<td>Number of geometric zones&lt;sup&gt;c&lt;/sup&gt; in the problem.</td>
</tr>
<tr>
<td>IM</td>
<td>Total number of coarse mesh intervals&lt;sup&gt;d&lt;/sup&gt; in the X (or R) direction.</td>
</tr>
<tr>
<td>IT</td>
<td>Total number of fine mesh intervals&lt;sup&gt;c&lt;/sup&gt; in the X (or R) direction.</td>
</tr>
</tbody>
</table>

---

<sup>a</sup> If SLAB2ANG is entered, the problem will be run as a 2-angle plane calculation and the Block-V parameter 12ANG need not be entered.

<sup>b</sup> Material is defined on page 2-41.

<sup>c</sup> Zone is defined on page 7-13.

<sup>d</sup> Coarse mesh is defined on page 7-13.

<sup>e</sup> Fine mesh is defined on page 7-13.
Storage Requirements
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSCM</td>
<td>Length of SCM desired (default=40000)</td>
</tr>
<tr>
<td>MAXLCM</td>
<td>Length of LCM desired (default=14000)</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINIPRT</td>
<td>Provide a shortened output print file. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>

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}*  

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSOLV</td>
<td>Suppress solver module execution. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOEDIT</td>
<td>Suppress edit module execution. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOGEOD</td>
<td>Suppress writing GEODST file even though the geometry input (Block-II) may be present. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOMIX</td>
<td>Suppress writing mixing files even though the mixing input in Block-IV may be present. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOASG</td>
<td>Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOMACR</td>
<td>Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOSLNP</td>
<td>Suppress writing the SOLINP file even though Block-V may be present. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOEDTT</td>
<td>Suppress writing the EDITIT file even though Block-VI may be present. $0/I = \text{no/yes}$</td>
</tr>
<tr>
<td>NOADJM</td>
<td>Suppress writing the ADJMAC file even though an adjoint calculation is called for. $0/I = \text{no/yes}$</td>
</tr>
</tbody>
</table>

Note: Default on all these controls is no.
Geometry Arrays

{Required}

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

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIB</strong></td>
<td>Name(^a) and form of the cross-section data file. Enter as a data item one of the following words:</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>ISOTXS(^b)</strong></td>
<td>CCCC standard isotope ordered binary cross-section file.</td>
</tr>
<tr>
<td><strong>XSLIB</strong></td>
<td>ASCII text library supplied in a separate file named XSLIB.</td>
</tr>
<tr>
<td><strong>ODNINP</strong></td>
<td>ASCII text library follows after this block of input (after the T of Block-III).</td>
</tr>
<tr>
<td><strong>GRUPXS(^c)</strong></td>
<td>CCCC standard group ordered cross-section file.</td>
</tr>
<tr>
<td><strong>BXSLIB</strong></td>
<td>Binary library supplied as a separate file named BXSLIB. [See “Binary Form of Card-Image Libraries (the BXSLIB file)” on page 10-12.</td>
</tr>
<tr>
<td><strong>MACRXS(^d)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>XSLIBB</strong></td>
<td>See “XSLIBB Card-Image Library File” on page 10-12.</td>
</tr>
<tr>
<td><strong>MACBCD</strong></td>
<td>ASCII form of MACRXS file.</td>
</tr>
<tr>
<td><strong>MENDF</strong></td>
<td>(LANL only) See “The Los Alamos MENDF5 Cross-Section Library” on page 10-13.</td>
</tr>
<tr>
<td><strong>MENDFG</strong></td>
<td>(LANL only) See “The Los Alamos MENDF5G Gamma Cross-Section Library” on page 10-14.</td>
</tr>
<tr>
<td><strong>other</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
## Nuclear Data Type and Options (Cont.)
### {Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>WRITMXS</td>
<td>Controls the code's writing certain ASCII cross-section files. Enter one of the following words:</td>
</tr>
<tr>
<td>{optional}</td>
<td>Word</td>
</tr>
<tr>
<td></td>
<td>MACBCD</td>
</tr>
<tr>
<td></td>
<td>XSLIBB</td>
</tr>
<tr>
<td></td>
<td>XSLIBE</td>
</tr>
<tr>
<td></td>
<td>XSLIBF</td>
</tr>
<tr>
<td>LNG</td>
<td>Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.</td>
</tr>
<tr>
<td>{optional}</td>
<td>BALXS</td>
</tr>
<tr>
<td></td>
<td>{optional}</td>
</tr>
<tr>
<td></td>
<td>WARNING See page 10-21 before using!</td>
</tr>
<tr>
<td></td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NTICHI</td>
<td>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.</td>
</tr>
<tr>
<td>{optional}</td>
<td>CHIVEC</td>
</tr>
</tbody>
</table>

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 12LPl=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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBNAME</td>
<td>Alternate name of the library file. May be used only with certain types of libraries. See Table 2.1.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>LIB</th>
<th>LIBNAME AVAILABLE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>No</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>Yes</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>Yes</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>ODNINP</td>
<td>No</td>
</tr>
<tr>
<td>MACBCD</td>
<td>No</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>No</td>
</tr>
<tr>
<td>MENDFA</td>
<td>No</td>
</tr>
<tr>
<td>MENDFGB</td>
<td>No</td>
</tr>
<tr>
<td>XSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>other</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXORD</td>
<td>Highest Legendre order in the scattering tables.</td>
</tr>
<tr>
<td>IHM</td>
<td>Number of positions (entries) in each row of the cross-section table.</td>
</tr>
<tr>
<td>IHT</td>
<td>Position number of the total cross section.</td>
</tr>
<tr>
<td>IHS</td>
<td>Position number of the self-scatter cross section.</td>
</tr>
<tr>
<td>IFIDO</td>
<td>Format of the cross-section library.</td>
</tr>
<tr>
<td>I2LP1</td>
<td>Higher order scattering cross sections on the library contain the 2L+1 term.</td>
</tr>
<tr>
<td>SAVBXS</td>
<td>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.</td>
</tr>
<tr>
<td>KWIKRD</td>
<td>Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? 0/1 = no/yes (default=yes).</td>
</tr>
<tr>
<td>NAMES [NISO]</td>
<td>Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).</td>
</tr>
<tr>
<td>EDNAME [IHT-3]</td>
<td>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.).</td>
</tr>
<tr>
<td>NTPI [NISO]</td>
<td>Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).</td>
</tr>
<tr>
<td>VEL [NGROUP]</td>
<td>Speeds for each group. Needed only for alpha calculations.</td>
</tr>
<tr>
<td>EBOUND [NGROUP+1]</td>
<td>Energy group boundaries.</td>
</tr>
</tbody>
</table>
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 self-scattering cross sections. Order within a row (e.g., for group g) is then as follows:

\[ \cdots \sigma_{\text{abs}}, \nu\sigma_f, \sigma_{\text{total}}, \cdots, \sigma_{g \rightarrow g}, \sigma_{g+1 \rightarrow g}, \sigma_{g-1 \rightarrow g}, \sigma_{g \rightarrow 2g}, \text{ etc.} \]

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 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 $\Sigma_g$ 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 $\Sigma_g$ 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLS(^a) [-;MT]</td>
<td>Instructions for mixing “isotopes” or premixes into “materials.” See details below.</td>
</tr>
<tr>
<td>ASSIGN(^b) [-;NZONE]</td>
<td>Assignments of materials to geometric zones. See below.</td>
</tr>
<tr>
<td>PREMIX [-;-]</td>
<td>Instructions for mixing “isotopes” into premixes. See below.</td>
</tr>
</tbody>
</table>

\(^a\) 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:

\[
\text{MATLS} = \text{mat}_1 \, \text{comp}_1 \, \text{den}_1, \ \text{comp}_2 \, \text{den}_2, \ ...\text{etc}.... ;
\]

where \(\text{mat}_1\) is the desired character name of the first material and \(\text{comp}_1, \text{comp}_2, \) and so on are the character names of its components which have “densities” of, respectively, \(\text{den}_1, \text{den}_2\), and so on. Additional materials (i.e., \(\text{mat}_2, \text{mat}_3, \) 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 \( \text{den}_i \) is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 2-47).

Short form: \( \text{MATLS}= \text{ISOS} \)

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:

\[
\text{MATLS}= \text{mat}_1 \text{ comp}_1; \quad \text{mat}_2 \text{ comp}_2 \text{ den}_2; \quad \text{...etc.}
\]

**ASSIGN input array**

The general form of the ASSIGN instruction is shown below:

\[
\text{ASSIGN}= \text{zone}_1 \text{ mat}_1 \text{ vol}_1, \text{ mat}_2 \text{ vol}_2, \text{...etc.}
\]

where \( \text{zone}_1 \) is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). \( \text{mat}_1, \text{mat}_2, \) and so on are the character names of the materials that will be present in this zone with, respectively, the “volume fractions” \( \text{vol}_1, \text{vol}_2, \) and so on. Additional zones (i.e., \( \text{zone}_2, \text{zone}_3, \) 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: \( \text{ASSIGN}= \text{MATLS} \)

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 \( \text{ASSIGN}= \text{MATLS} \) 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:

\[
\text{PREMIX}= \text{mat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \text{...etc.}
\]
where \( \text{tmat}_1 \) is the character name of the first material and \( \text{comp}_1, \text{comp}_2, \) and so on are the character names of its components which have "densities" of, respectively, \( \text{den}_1, \text{den}_2, \) and so on. Additional temporary materials (i.e., \( \text{tmat}_2, \text{tmat}_3, \) 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}*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASGMOD(^a) [-;-]</td>
<td>( C_1 ) parameters used in concentration searches. See the discussion below.</td>
</tr>
</tbody>
</table>

\(^a\) 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:

\[ C(z,x) = C_0(z,x) + C_1(z,x) \times CMOD \]

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:

\[ \text{ASGMOD} = \text{zone mat}_m \text{ vol}_m \text{ mat}_n \text{ vol}_n \text{ ...etc... ;} \]

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOD</td>
<td>Concentration modifier. Input value is not used in a search. See the discussion below.</td>
</tr>
</tbody>
</table>
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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATNAM [MT]</td>
<td>Character material names for Materials. Used only if the mat$^i$ 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.</td>
</tr>
<tr>
<td>ZONNAM [NZONE]</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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  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]| 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:  
  \[ \text{ATWT} = iso_1 \, atwt_1 \quad iso_2 \, atwt_2 \quad \ldots \]  
  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

<table>
<thead>
<tr>
<th>Name</th>
<th>Calculation type: Enter one of the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEVT</td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td>ISCT</td>
<td>Legendre order of scattering (default=0).</td>
</tr>
<tr>
<td>ITH</td>
<td>Calculation Mode. 0/1 = direct/adjoint calculation (default=0).</td>
</tr>
<tr>
<td>IBL</td>
<td>Left boundary condition$^a$. 0/1/2/3 = vacuum/reflective$^b$/periodic/white$^c$ (default=vacuum).</td>
</tr>
<tr>
<td>IBR</td>
<td>Right boundary condition: Same input values as for IBL.</td>
</tr>
</tbody>
</table>

---

$^a$ 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPSI</td>
<td>Inner iteration convergence precision (default=0.0001).</td>
</tr>
<tr>
<td>EPSO</td>
<td>Outer iteration convergence precision (default=EPSI).</td>
</tr>
<tr>
<td>IITL</td>
<td>Maximum number of inner iterations per group until $1.0-\lambda &lt; 3\cdot$EPSO. Recommend default be used. (default is chosen by code).</td>
</tr>
<tr>
<td>IITM</td>
<td>Maximum number of inners per group allowed after $</td>
</tr>
<tr>
<td>OITM</td>
<td>Maximum number of outer iterations (default=20).</td>
</tr>
<tr>
<td>ITLIM</td>
<td>Number of seconds time limit (default=unlimited).</td>
</tr>
</tbody>
</table>

## Acceleration Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREYACC</td>
<td>Upscatter acceleration$^a$. $0/1 = no/yes^b$.</td>
</tr>
</tbody>
</table>

---

*a.* See page 7-21 for details.

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

K-Code Convergence
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCALC</td>
<td>Special Criticality Convergence Scheme. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>

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}

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

### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRCOR</td>
<td>Apply transport correction(^a) to cross sections on MACRXS file. Enter one of the following words:</td>
</tr>
<tr>
<td></td>
<td><strong>Word</strong></td>
</tr>
<tr>
<td></td>
<td><strong>DIAG</strong></td>
</tr>
<tr>
<td></td>
<td><strong>BHS</strong></td>
</tr>
<tr>
<td></td>
<td><strong>CESARO</strong></td>
</tr>
<tr>
<td></td>
<td><strong>NO</strong></td>
</tr>
<tr>
<td>BHGT</td>
<td>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).</td>
</tr>
<tr>
<td>BWTH</td>
<td>Buckling width. Used only for plane and two-angle plane geometries. (default = 0.0, which is treated as infinity).</td>
</tr>
<tr>
<td>NORM</td>
<td>Normalization constant. Normalize the fission source rate to this value when IEVT (\geq 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_{\text{eff}}) problems where the integral is equal to NORM*(k_{\text{eff}}).) Any fluxes printed by setting FLUXP nonzero will be normalized consistently with this source rate.</td>
</tr>
<tr>
<td>I2ANG</td>
<td>Do two-angle plane calculation? 0/1 = no/yes. For IGEOM=1 only.</td>
</tr>
<tr>
<td>CHI ([\text{NGROUP};M])</td>
<td>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.</td>
</tr>
<tr>
<td>DEN ([\text{IT}])</td>
<td>Density factor to use in each fine mesh interval. Applied to the zone macroscopic cross sections at each mesh interval.</td>
</tr>
</tbody>
</table>

---

\(^a\) 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQUAD</td>
<td>Source of quadrature constants. Enter one of the following:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-3</td>
<td>Get constants from SNCONS file.</td>
</tr>
<tr>
<td>1</td>
<td>Traditional built-in $P_n$ constants $[S_2, S_4, S_6, S_8, S_{12}, S_{16}, \ldots]$ (This is the default).</td>
</tr>
<tr>
<td>2</td>
<td>Traditional built-in $D_{P_n}$ constants $[S_4, S_8, S_{12}, S_{16}, S_{24}, S_{32}, S_{40}, S_{48}, S_{64}, \text{ or } S_{96}]$.</td>
</tr>
<tr>
<td>4</td>
<td>Built in $GQ_n$ set (recommended for cylinders) $[S_2, S_4, S_6, S_8, S_{12}, \text{ or } S_{16}]$.</td>
</tr>
<tr>
<td>WGT\textsuperscript{a} [MM\textsuperscript{b}] {optional}</td>
<td>Quadrature weights.</td>
</tr>
<tr>
<td>MU [MM]  {optional}</td>
<td>Mu cosines.</td>
</tr>
</tbody>
</table>

\textsuperscript{a} Presence of the WGT and MU arrays overrides the IQUAD input

\textsuperscript{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\}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFLUX</td>
<td>Read initial flux guess from the RTFLUX file. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>
General Eigenvalue Search Control\(a\)
\{ IEVT >1\} 

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

\(a\). 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 \(k_{\text{eff}}\) or \(\alpha\) 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\}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM [IM]</td>
<td>Radius modifiers for each coarse mesh interval, for use with</td>
</tr>
<tr>
<td></td>
<td>dimension searches only.</td>
</tr>
</tbody>
</table>

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:
ONEDANT INPUT DETAILS

Block-V Details: Solver Input

\[ X_{MESH_{i+1}} = X_{MESH_i} + \{X_{MESH_{i+1}} - X_{MESH_i}\} \times [1.0 + EV \times RM_i], \quad i = 1, \ldots, IM \]

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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 array in Block-IV.</td>
</tr>
</tbody>
</table>

\(a.\) 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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSORS</td>
<td>Read source from interface file FIXSRC. 0/1 = no/yes.</td>
</tr>
<tr>
<td>SOURCE [NGROUP; NMQ(^a)]</td>
<td>Source spectrum for each of NMQ moments. (Spatial distribution is assumed to be flat with value unity)</td>
</tr>
<tr>
<td>SOURCXX [IT;NMQ]</td>
<td>X (or R) spatial distribution for each moment. (Spectrum is assumed to be flat with value unity)</td>
</tr>
<tr>
<td>SOURCE [NGROUP; NMQ]</td>
<td>Source spectrum.</td>
</tr>
<tr>
<td>SOURCXX [IT;NMQ]</td>
<td>X (or R) spatial distribution for each moment.</td>
</tr>
<tr>
<td>SOURCFF [IT;NGROUP*NMQ]</td>
<td>Spatial distribution for each row, group, and moment.</td>
</tr>
</tbody>
</table>

---

**Notes:**

- **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 angle plane. \(n\) must be less than or equal to ISCT. See page 12-24 for more details.

- The full source at mesh point \(i\) in group \(g\) for moment \(m\) is calculated as follows:
  
  \[ \text{SOURC}(i,g,m) = \text{SOURCE}(g,m)\times\text{SOURCXX}(i,m) \]
## Boundary Source Input

### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SILEFT [NGROUP]</td>
<td>Isotropic source on the left side in each energy group. For plane geometry only!</td>
</tr>
<tr>
<td>SIRITE [NGROUP]</td>
<td>Isotropic source on the right side in each energy group.</td>
</tr>
</tbody>
</table>

### Option 1: Isotropic Boundary Source.

### Option 2: Anisotropic Boundary Source.

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALEFT [MM/2^a;NGROUP]</td>
<td>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.</td>
</tr>
<tr>
<td>SARITE [MM/2;NGROUP]</td>
<td>Angular fluxes on the right side.</td>
</tr>
</tbody>
</table>

---

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBEDO [NGROUP]</td>
<td>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.</td>
</tr>
<tr>
<td>RBEDO [NGROUP]</td>
<td>Right boundary albedoes for each group for all geometries. Applied as albedoes for either reflective (IBR=1) or white (IBR=3) boundary conditions.</td>
</tr>
</tbody>
</table>
Block-VI Details: Edit Input*

Spatial Specifications for Reaction Rates
{Required^a}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTED</td>
<td>Do edits by fine mesh. 0/1 = no/yes.</td>
</tr>
<tr>
<td>ZNED</td>
<td>Do edits by zone. 0/1 = no/yes. (i.e., edit zone, not SOLVER zone.</td>
</tr>
<tr>
<td>POINTS [≤IT]</td>
<td>Fine mesh point (or interval) numbers at which point edits are desired.</td>
</tr>
<tr>
<td>{optional}</td>
<td>USED ONLY IF PTED=1. (default= all mesh intervals)</td>
</tr>
<tr>
<td>EDZONE [IT]</td>
<td>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).</td>
</tr>
<tr>
<td>{optional}</td>
<td></td>
</tr>
</tbody>
</table>

^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**

{Optional

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDXS [≤NEDT]</td>
<td>Cross-section types to be used in forming reaction rates.</td>
</tr>
<tr>
<td>(required(^c))</td>
<td>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.</td>
</tr>
<tr>
<td>RESDNT</td>
<td>Do edits using the resident macroscopic cross section at each point. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDISOS [≤NISO]</td>
<td>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).</td>
</tr>
<tr>
<td>EDCONS [≤NISO]</td>
<td>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).</td>
</tr>
<tr>
<td>EDMATS [≤MT]</td>
<td>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).</td>
</tr>
<tr>
<td>XDF(^d) [IT]</td>
<td>Fine mesh density factors for the x (or r) directions.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CROSS-SECTION INPUT VIA ISOTXS or GRUPXS</th>
<th>CROSS-SECTION INPUT VIA ASCII TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>EDIT</strong> Position</td>
</tr>
<tr>
<td>chi</td>
<td>1</td>
</tr>
<tr>
<td>nu-fission</td>
<td>2</td>
</tr>
<tr>
<td>total</td>
<td>3</td>
</tr>
<tr>
<td>absorption</td>
<td>4</td>
</tr>
<tr>
<td>(n,p)</td>
<td>5</td>
</tr>
<tr>
<td>(n,d)</td>
<td>6</td>
</tr>
<tr>
<td>(n,t)</td>
<td>7</td>
</tr>
<tr>
<td>(n,alpha)</td>
<td>8</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>9</td>
</tr>
<tr>
<td>(n,gamma)</td>
<td>10</td>
</tr>
<tr>
<td>fission</td>
<td>11</td>
</tr>
<tr>
<td>transport</td>
<td>12</td>
</tr>
</tbody>
</table>

<sup>a</sup> Names are eight characters. A period within a name in this table denotes a blank.

<sup>b</sup> 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.

<sup>c</sup> 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]}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSFE [NGROUP;M]</td>
<td>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.</td>
</tr>
<tr>
<td>RSFX [IT;M]</td>
<td>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).</td>
</tr>
<tr>
<td>RSFNAM [M]</td>
<td>Character names for the M user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM).</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOLL [NBG]</td>
<td>Edit energy group collapsing option:</td>
</tr>
<tr>
<td></td>
<td>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)</td>
</tr>
<tr>
<td>IGRPED</td>
<td>Print option on energy groups. Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
### Reaction Rate Summing

**{Optional}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICSUM [&lt;500 sums]</td>
<td>Cross-section reaction rate summing specifications.</td>
</tr>
</tbody>
</table>

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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSED</td>
<td>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.</td>
</tr>
</tbody>
</table>
## Power Normalization

**{Optional}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>Normalize to POWER megawatts.(^a)</td>
</tr>
<tr>
<td>{required}</td>
<td>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.</td>
</tr>
<tr>
<td>MEVPER</td>
<td>MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power.</td>
</tr>
<tr>
<td>{required}</td>
<td>Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 2-51).</td>
</tr>
</tbody>
</table>

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

---

---

---
### Miscellaneous Edit Items

**{Optional}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RZFLUX</td>
<td>Write the CCCC standard zone flux file RZFLUX or AZFLUX. 0/1 = no/yes.</td>
</tr>
<tr>
<td>RZMFLX</td>
<td>Write the code-dependent zone flux moments file RZMFLX or AZMFLX. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDOUTF(^c)</td>
<td>ASCII output files control. Enter one of the following values:</td>
</tr>
<tr>
<td><strong>Value</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-3</td>
<td>Write both EDTOGX (without scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>-2</td>
<td>Write EDTOGX file (without scalar fluxes).</td>
</tr>
<tr>
<td>0</td>
<td>Write neither file. (default)</td>
</tr>
<tr>
<td>1</td>
<td>Write EDTOUT file.</td>
</tr>
<tr>
<td>2</td>
<td>Write EDTOGX file (with scalar fluxes).</td>
</tr>
<tr>
<td>3</td>
<td>Write both EDTOGX (with scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>BYVOLP</td>
<td>Printed point reaction rates will have been multiplied by the mesh volume. 0/1 = no/yes.</td>
</tr>
<tr>
<td>AJED(^d)</td>
<td>Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. 0/1 = regular/adjoint.</td>
</tr>
<tr>
<td>FLUXONE</td>
<td>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.</td>
</tr>
</tbody>
</table>

---

\(a\). 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

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi$</td>
<td>CHI</td>
<td>fission spectrum</td>
</tr>
<tr>
<td>$\nu\sigma_f$</td>
<td>NUSIGF</td>
<td>effective nu-sigma-fission</td>
</tr>
<tr>
<td>$\sigma_t$</td>
<td>TOTAL</td>
<td>Total cross section</td>
</tr>
<tr>
<td>$\sigma_a$</td>
<td>ABS</td>
<td>absorption (^a)</td>
</tr>
<tr>
<td>(n,n)</td>
<td>MEND1</td>
<td>elastic scattering</td>
</tr>
<tr>
<td>(n,n')</td>
<td>MEND2</td>
<td>inelastic scattering</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>MEND3</td>
<td>n,2n scattering</td>
</tr>
<tr>
<td>(n,3n)</td>
<td>MEND4</td>
<td>n,3n scattering</td>
</tr>
<tr>
<td>(n,$\gamma$)</td>
<td>MEND5</td>
<td>gamma production</td>
</tr>
<tr>
<td>(n,$\alpha$)</td>
<td>MEND6</td>
<td>alpha production</td>
</tr>
<tr>
<td>(n,p)</td>
<td>MEND7</td>
<td>proton production</td>
</tr>
<tr>
<td>(n,f)</td>
<td>MEND8</td>
<td>direct fission</td>
</tr>
<tr>
<td>(n,n')f</td>
<td>MEND9</td>
<td>second-chance fission</td>
</tr>
<tr>
<td>(n,2n)f</td>
<td>MEND10</td>
<td>third-chance fission</td>
</tr>
<tr>
<td>(n,F)</td>
<td>N-FISS</td>
<td>$[(n,F) = (n,f) + (n,n')f + (n,2n)f]$</td>
</tr>
<tr>
<td>$\chi_p$</td>
<td>MEND12</td>
<td>prompt fission spectrum (only for fissionable materials)</td>
</tr>
<tr>
<td>$\chi_t$</td>
<td>MEND13</td>
<td>total fission spectrum (only for fissionable materials)</td>
</tr>
</tbody>
</table>

\(^a\) $\sigma_a$ for group $g$ is defined as $\sigma_a = \sigma_t - \sum_{g' \rightarrow g} \sigma_{g' \rightarrow g'}$.

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


APPENDIX A: SAMPLE INPUT

This appendix presents the printed output from each of two sample problems. The first sample problem is a standard $k_{\text{eff}}$ 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 $k_{\text{eff}}$ Calculation

Sample Problem 1 is a standard $k_{\text{eff}}$ 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, \((\text{U-238, PU-239})\text{O}_2\), 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 S4 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 S4 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 \(\xi\)-level index. The terms BETA PLUS and BETA MINUS refer, respectively to the terms...
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard $k_{eff}$ Calculation

$\alpha_{m+1/2}/w_m$ 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 $\chi$ (fission fraction), $v\sigma_f$, $\sigma_p$, and $\sigma_a$. The scattering
matrix terms correspond to the coefficients in a Legendre expansion of the term
$\sigma_{s,g'(r, \mu_0)}$ 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) $\text{SOURCE} = \text{total inhomogeneous source} = QG_g$

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

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 macro-
scopic cross section.
interval i, \(A_{l+1/2}\) 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,

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

For a source with fission problem,

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

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

\[
SIN_g = \sum_{h=1}^{\text{NGROUP}} \sum_{i=1}^{\text{IT}} (\sigma_{s,h \rightarrow g})_{i} \phi_{h,i} V_i;
\]

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

\[
SS_g = \sum_{i=1}^{\text{IT}} (\sigma_{s,g \rightarrow g})_{i} \phi_{g,i} V_i;
\]

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

\[
SOUT_g = \sum_{i=1}^{\text{IT}} (\sigma_{t,g})_{i} \phi_{g,i} V_i - ABG_g - SS_g;
\]

where \(\sigma_{t,g}\) is the total cross section for group \(g\) plus any buckling "absorption" plus any "time absorption" \(\left(\alpha/v_g\right)\);

(6) ABSORPTION = absorption in group \(g\) = \(ABG_g\)

\[
ABG_g = \sum_{i=1}^{\text{IT}} (\sigma_{a,g})_{i} \phi_{g,i} V_i;
\]

where \(\sigma_{a,g}\) is the absorption cross section for group \(g\) plus any buckling "absorption" plus any "time absorption" \(\left(\alpha/v_g\right)\);
Sample Problem 1: Standard k\text{eff} Calculation

(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 = \( BAL_g \)

\[
BAL_g = 1 - \frac{NL_g + ABG_g + SOUT_g}{OG_g + FG_g + SIN_g};
\]

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

\[
NPROD_g = \frac{1}{\sum_{i=1}^{IT}} \sum_{g=1}^{NGROUP} (v\sigma_f)_{g,i} \phi_{g,i} V_i;
\]

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

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 \((v\sigma_f)_{g,i}\phi_{g,i}V_i\), 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.
APPENDIX A: SAMPLE INPUT
Sample Problem 1: Standard keff Calculation

---

List of cards in the input stream...

1.  2 0 0
2. sample problem 1 for user's manual
3. standard keff 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 (d.f) format
7. mixing - isotopes mixed to make materials named steel, fuel, b nickel, and sodium
8. materials assigned to make zones named core, blanket, and shield
9. solve - card input supplied
10. edits - none
11. block 1
12. igcm=2, nrgroup=2, is=4 nuce=7 st=4 zone=3 lts=50 t
13. block 2
14. block 3 (cross sections)***
15. lib costing
16. named load level 6 lib=4 load=5 ifidobj initial
17. names = "o-16" "u-235" "u-238" "pu-239" "u-238" end
18. edenc= files
19.
20. since lib costing, the cross-section library in card-images
21. will begin immediately following the block III terminal "*
22. note that a title card precedes each cross-section
23. block (since initial)**
24. t
25. oxygen-16 (o-18) sample 2 group flux cross sections
26. 0.000 0.010 0.000 2.000 1.000 0.000 o18/1
27. 0.000 0.050 0.000 3.000 3.000 0.390 o18/2
28. sodium (neq3) sample 2 group flux cross sections
29. 0.000 0.030 0.000 1.200 3.500 0.000 neq3/1
30. 0.000 0.055 0.000 4.000 3.950 0.388 neq3/2
31. iron (fe) sample 2 group flux cross sections
32. 0.000 0.010 0.000 2.000 1.700 0.000 fe/1
33. 0.000 0.015 0.000 4.000 4.900 0.392 fe/2
34. chromium (cr) sample 2 group flux cross sections
35. 0.000 0.013 0.000 2.450 2.150 0.000 cr/1
36. 0.000 0.020 0.000 5.000 4.980 0.297 cr/2
37. nickel (ni) sample 2 group flux cross sections
38. 0.000 0.010 0.000 2.400 2.000 0.000 ni/1
39. 0.000 0.015 0.000 3.000 2.700 0.330 ni/2
40. plutonium (pu-239) sample 2 group flux cross sections
41. 1.000 1.250 4.800 2.000 0.000 pu239/1
42. 1.000 2.500 4.000 12.000 9.500 0.650 pu239/2
43. uranium (u-238) sample 2 group flux cross sections
44. 0.300 0.400 0.900 4.700 3.000 0.000 u238/1
45. 0.300 0.500 0.900 13.000 12.500 1.300 u238/2
46. end of cross-section data
47. note that there is no terminal "*" since the cross sections are
48. in los alamos (d.f) format (ifidobj=0****
49. block iv (mixing)****
50. matlist: steel, fe ,0; cr, 0.01; sodium , 0.02; blc (u-235), 0.055; "o-16", 0.0412;
51. sodium , 0.283; 0.0008, "o-15", 0.0412;
52. assign core dull, 0.25, sodium , 0.25; blanket blc, 0.35, sodium , 0.25;
53. shield: sodium , 0.7, steel , 0.3 t
54. block v (solver)**
55. lev=1 iter=0 ibid=0
56. flagged 1 flagged2 flagged3
57. chi(0.6,0.4; 0.7, 0.3 t
58. no information supplied

---

2-74

ONEDANT USER'S GUIDE Version 3.0
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard keff Calculation

```
case title

**has not been used

**

*key start case input*

**

2 read  number of title cards to follow
0 noisy  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 keff calculation, all input by means of card images

*key end  block i read*

**

...block i - controls and dimension...

**

...dimensions (array name = dimen)...

2 igcm  1/2/3  place/cylinder/sphere
2 ngcm  number of energy groups
4 ign  angular quadrature order
7 niso  number of input isotopes (from isotos, group, or cards)
4 mnc  number of permanent materials
3 nzone  number of zones
3 nme  number of coarse mesh x intervals
50 it  number of fine mesh intervals

...storage...

maxcm  140000
maxcm  40000

*key end  block iii read-es*

**

APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard keff Calculation

```

ONEDANT USER'S GUIDE  Version 3.0  2-75
Sample Problem 1: Standard k_{eff} Calculation

---

**APPENDIX A: SAMPLE INPUT**

### Sample Problem 1: Standard k_{eff} Calculation

---

---

#### Block III - Cross-section Library...

---

#### Library Source...

---

#### Card Library Parameters (array name = cards)...

---

<table>
<thead>
<tr>
<th>Group</th>
<th>Chi</th>
<th>Vel</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Group</th>
<th>Chi</th>
<th>Vel</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>2</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

Last neutron group (Inh) is number 2

---

#### Energy Structure...

---

**Key start card lib read**

---

#### Header Cards from the Card Library...

---

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Isotope</th>
<th>Number</th>
<th>Name</th>
<th>Order</th>
<th>Header Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>16-oxygen</td>
<td>0-16</td>
<td>oxygen-16</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
<tr>
<td>2.</td>
<td>23-neutron</td>
<td>0-23</td>
<td>sodium-23</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
<tr>
<td>3.</td>
<td>22-calcium</td>
<td>0-22</td>
<td>iron-22</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
<tr>
<td>4.</td>
<td>12-calcium</td>
<td>0-12</td>
<td>chromium-12</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
<tr>
<td>5.</td>
<td>57-nickel</td>
<td>0-57</td>
<td>nickel-57</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
<tr>
<td>6.</td>
<td>239-plutonium</td>
<td>0-239</td>
<td>plutonium-239</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
<tr>
<td>7.</td>
<td>238-plutonium</td>
<td>0-238</td>
<td>uranium-238</td>
<td>sample 2</td>
<td>group before cross sections</td>
</tr>
</tbody>
</table>

---

**Key end card lib read**

---

**Key end block iv read-ends**

---

---

---
### Sample Problem 1: Standard $k_{eff}$ Calculation

<table>
<thead>
<tr>
<th>mix</th>
<th>comp</th>
<th>density</th>
<th>comp</th>
<th>density etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fe</td>
<td>1</td>
<td>5.00E-02</td>
<td>Cr</td>
<td>1.60E-02, Ni 1.00E-02</td>
</tr>
<tr>
<td>U-238</td>
<td>2</td>
<td>5.10E-03</td>
<td>U-235</td>
<td>1.55E-02, O-16 4.12E-02,</td>
</tr>
<tr>
<td>Na-23</td>
<td>3</td>
<td>2.06E-02</td>
<td>O-16</td>
<td>4.12E-02,</td>
</tr>
<tr>
<td>Na-23</td>
<td>4</td>
<td>2.50E-02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- Assign
- Core fuel 3.50E-02, sodium 4.00E-01, steel 2.50E-02
- Blank lead 3.50E-01, sodium 4.00E-01, steel 2.50E-01
- Shield sodium 7.00E-01, steel 3.00E-01

```plaintext
*key start mix card xe *

*key end mix card xe *
```

---

**Key:**
- *key start mix card xe*:
- *key end mix card xe*
- *key end block v real-solve*
- *key end input module*
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard k-eff Calculation

---

Sample problem 1 for user's manual.

---

...required input (array name = salin)...

<table>
<thead>
<tr>
<th>Raw Input</th>
<th>Defaulted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>lev = -1/0/1/2/3/4 - type of calculation</td>
</tr>
<tr>
<td></td>
<td>1 homogeneous source with fission and/or upscatter</td>
</tr>
<tr>
<td></td>
<td>0 homogeneous source alone</td>
</tr>
<tr>
<td></td>
<td>2 k-effective</td>
</tr>
<tr>
<td></td>
<td>2 alpha or time absorption search</td>
</tr>
<tr>
<td></td>
<td>3 concentration search</td>
</tr>
<tr>
<td></td>
<td>4 delta (i.e., dimension) search</td>
</tr>
<tr>
<td>0 0</td>
<td>levt = legendre order of scattering</td>
</tr>
<tr>
<td>0 1</td>
<td>ith = 0/1 - direct/adjoint - mode of calculation (default = direct)</td>
</tr>
<tr>
<td>0 1</td>
<td>ibr = 0/1/2/3 - left boundary condition</td>
</tr>
<tr>
<td></td>
<td>vacuum/reflective/periodic/white</td>
</tr>
<tr>
<td>0 0</td>
<td>ibr = 0/1/2/3 - right boundary condition</td>
</tr>
<tr>
<td></td>
<td>vacuum/reflective/periodic/white</td>
</tr>
</tbody>
</table>

---

...convergence controls (array name = iter)...

- 0.0005/0.0008-04 epsilon inner iteration convergence criterion (default = 0.0001)
- 0.0005/0.0008-04 epsilon outer iteration convergence criterion (default = epsilon)
- 0 1 itil medium number of inner iterations per group until (1-lambda)lt.l*epsilon (default calculated)
- 0 20 ilim maximum number of outer iterations per group (default = 50)
- 0 0 ialim 0/1, np/yso, only converge the eigenvalue (to 0.001)
- 0 0 itlim iteration time limit (seconds)

---

this sample problem run on 8/25/94 with solver version 08-04-944 product release 2.4m machine turkler
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard keff Calculation

...block v -- solver input (continued)...

raw as input detailed

...miscellaneous parameters(array name = misc)...

0.003E+00 0.002E+00
0.002E+00 0.002E+00
0.008E+00 0.008E+00
0.008E+00 0.008E+00

bleq: buckling height
bwcl: buckling width(plane geom only)

0 0
0 0
0 0
0 1

influx 0/1 no/yes - read input flux from file rflux (at flux for adjoint)
imvec 0/1 no/yes - read input source from file isvec
ilnvec 0/1 no/yes - do 2 angle calculation
lged -3/1/2/3 - source of geometry constants (default=1)
file geom/built in p-n/built in dp-e cards

...output controls(array name = solout)...

1 fluxp 0/1/2 none/isotropic/all elements - flux print
2 specp 0/1/2 none/principal/all - macroscopic cross-section print
1 fluxp 0/1 no/yes - print flux solution rate
0 soureg 0/1/2/3 no/yes read/normalized/zoom - print inhomogeneous source
0 geom 0/1 no/yes - print fine mesh geometry
0 emap 0/1 no/yes - print angular fluxes
0 rflux 0/1 no/yes - write angular fluxes to file rflux (at flux if adjoint)
0 mflux 0/1 no/yes - write flux moments to file mflux
0 halp 0/1 no/yes - print coarse mesh balances

...parameters inferred from input arrays...

2 inchi 0/1/2 none/one chi/nonewise chi
0 iden 0/1 no/yes - use density factors
0 ign 0/1 no/yes - use ign source density
0 lic0 number of source vectors input
0 lic0 number of source vectors input
0 lic0 number of source vectors input
0 ilat -1/0/1 isotropic/n0/all angles - left boundary source
0 ilat -1/0/1 isotropic/n0/all angles - right boundary source
0 ilat 0/1 no/yes - there are alldecs

...parameters from block i...

2 ilucm 1/2/3 plane/cylinder/sphere
2 nrgup number of energy groups
4 ilen angular quadrature order
4 st number of present materials
3 nzone number of zones
3 lsa number of coarse mesh x intervals
50 it number of fine mesh x intervals

ONEDANT USER'S GUIDE   Version 3.0   2-79
**APPENDIX A: SAMPLE INPUT**

**Sample Problem 1: Standard k eff Calculation**

---

...material assignments to zones...

---

zone cross section = sum over mats in the zone of (matl cross section)*(c0 + c1*cmd) 

with 

cmd = 0.0000000E+00

table:

<table>
<thead>
<tr>
<th>entry</th>
<th>zone</th>
<th>material</th>
<th>c0</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>no.</td>
<td>name</td>
<td>no.</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>core</td>
<td>2</td>
<td>fuel</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>sodium</td>
<td>3</td>
<td>steel</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>steel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>blanket</td>
<td>3</td>
<td>blks</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>blanket</td>
<td>4</td>
<td>steel</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>blanket</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>shield</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>shield</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

...zone assignment map...

---

key start map

---

r 0. 00. 100.

r 20. 00. 00.

wxms 20 15 15

course mesh 1 2 3

...comments from oxidant on storage requirements...

---

key start storage note

---

storage required allowed

small core 1524 40000

large core 670 14000

---

angular quadrature data for oxidant...

---

xi-level starting cosine

1 3.003743E-01

2 9.460423E-01

nu-level level weight level cosine

1 1.720974E-01 8.611363E-01

2 1.260793E-01 3.39981E-01

3 1.260793E-01 3.39981E-01

4 1.720974E-01 8.611363E-01

m refli m li point weight nu cosine where mu beta plus beta minus xi phi(deg)

1 4 1 1.630363E-01 3.39981E-01 5.54292E-02 3.39981E-01 0.000000E+00 0.000000E+00 0.000000E+00 8.611363E-01 13.4715

2 6 2 1.720974E-01 8.611363E-01 1.47752E-01 8.611363E-01 0.000000E+00 0.339981E04 156.3027

3 5 2 1.630363E-01 3.39981E-01 5.54292E-02 3.39981E-01 0.000000E+00 0.339981E04 156.3027

4 3 1 1.630363E-01 3.39981E-01 5.54292E-02 3.39981E-01 0.000000E+00 0.339981E04 48.0298

5 3 2 1.630363E-01 3.39981E-01 5.54292E-02 3.39981E-01 0.000000E+00 0.339981E04 68.8027

6 2 2 1.720974E-01 8.611363E-01 1.47752E-01 0.000000E+00 8.611363E-01 0.339981E04 23.8973

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

---

2-80

ONEDANT USER'S GUIDE Version 3.0
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard keff Calculation

```plaintext
# Cross-section related data from file sucede 000000100193 version 1

... Nails available from file sucede...

1 steel 2 fuel 3 blkt 4 sodium

# Cross sections for legendre orders up to p6...

**Key start mac cross sections**

```

```
Sample Problem 1: Standard k_eff Calculation

...iteration controls and criteria...

***iteration criteria***

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>liml</td>
<td>inner iteration count</td>
<td>1</td>
<td>terminates inner</td>
</tr>
<tr>
<td>limh</td>
<td>inner iteration count</td>
<td>20</td>
<td>terminates inner</td>
</tr>
<tr>
<td>epal</td>
<td>fractional leakage flux change per inner</td>
<td>1.000E-04</td>
<td>does another inner</td>
</tr>
<tr>
<td>diff sub-outers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>outd</td>
<td>sub-outer iteration count</td>
<td>100</td>
<td>terminates sub-outers</td>
</tr>
<tr>
<td>epao</td>
<td>fractional leakage change per sub-outer</td>
<td>2.98E-04</td>
<td>does another sub-outer</td>
</tr>
<tr>
<td>epox</td>
<td>max fractional leakage flux change per sub-outer</td>
<td>0.20E-06</td>
<td>does another sub-outer</td>
</tr>
</tbody>
</table>

Final convergence criteria

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>oitm</td>
<td>outer iteration count</td>
<td>20</td>
<td>quits with error message</td>
</tr>
<tr>
<td>epos</td>
<td>leakage-1.0</td>
<td>1.000E-04</td>
<td>does another outer</td>
</tr>
<tr>
<td>epox</td>
<td>max fractional leakage flux change</td>
<td>2.98E-04</td>
<td>does another outer</td>
</tr>
</tbody>
</table>

...flux and eigenvalue convergence as monitored by onedant...

**KOA start iteration monitor**

<table>
<thead>
<tr>
<th>time</th>
<th>outer</th>
<th>diffusion</th>
<th>k-eff</th>
<th>lamda-1</th>
<th>max pdx</th>
<th>inners</th>
<th>Eigenvalue</th>
<th>lamda-1</th>
<th>lamda-1</th>
<th>lamda-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9.940698E-01</td>
<td>0.28743E-02</td>
<td>1</td>
<td>4.57915E-03</td>
<td>6.79135E-04</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>0.28</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>9.940698E-01</td>
<td>0.28743E-02</td>
<td>1</td>
<td>4.57915E-03</td>
<td>6.79135E-04</td>
<td>yes</td>
<td><strong>np</strong></td>
</tr>
<tr>
<td>0.33</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>9.940698E-01</td>
<td>0.28743E-02</td>
<td>1</td>
<td>4.57915E-03</td>
<td>6.79135E-04</td>
<td>yes</td>
<td><strong>np</strong></td>
</tr>
</tbody>
</table>

--- inner iteration summary for outer iteration no. 4 ---

user per max flux at

group group change mesh
1 3 0.576-04 47
2 2 0.876-04 50

$\text{SSSSSS all convergence criteria satisfied SSSSS}$

<table>
<thead>
<tr>
<th>time</th>
<th>outer</th>
<th>diffusion</th>
<th>k-eff</th>
<th>lamda-1</th>
<th>Flux change converged</th>
<th>max pdx</th>
<th>inners</th>
<th>Eigenvalue</th>
<th>lamda-1</th>
<th>lamda-1</th>
<th>lamda-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.40</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>9.940698E-01 -9.96732E-06</td>
<td>1.47951E-04</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

particle balance = -1.63E3E-06 total inners all outers = 9

$\text{SSSSSS all convergence criteria satisfied SSSSS}$
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard keff Calculation

******************************************************************************

...group edit and balances upon convergence...

******************************************************************************

...title—sample problem 1 for user's manual

...system balance tables... (neutrons only)

******************************************************************************

**Key start balance tables**

<table>
<thead>
<tr>
<th>gp</th>
<th>source</th>
<th>fission source</th>
<th>in scatter</th>
<th>self scatter</th>
<th>out scatter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000000</td>
<td>6.02451296-01</td>
<td>1.11022368-16</td>
<td>1.42276988-02</td>
<td>4.91572768-01</td>
</tr>
<tr>
<td>2</td>
<td>0.0000000000</td>
<td>3.97548700-01</td>
<td>4.91572768-01</td>
<td>3.92956710-01</td>
<td>7.10542746-15</td>
</tr>
<tr>
<td>total</td>
<td>0.0000000000</td>
<td>1.0000000000</td>
<td>4.91572768-01</td>
<td>4.14684225-01</td>
<td>4.91572768-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gp</th>
<th>absorption</th>
<th>particle balance</th>
<th>right leakage</th>
<th>net leakage</th>
<th>reprod spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.02712508-01</td>
<td>-1.49848514-07</td>
<td>2.16556920-03</td>
<td>2.16556920-03</td>
<td>2.24793328-01</td>
</tr>
<tr>
<td>2</td>
<td>8.12076486-01</td>
<td>-2.42106544-06</td>
<td>5.70425293-02</td>
<td>5.70425293-02</td>
<td>7.24208768-01</td>
</tr>
<tr>
<td>total</td>
<td>9.40787236-01</td>
<td>-1.61963485-06</td>
<td>5.92151828-02</td>
<td>5.92151828-02</td>
<td>1.00000000-00</td>
</tr>
</tbody>
</table>

******************************************************************************

**Integral summary information**

<table>
<thead>
<tr>
<th>summary</th>
<th>integral-keff</th>
<th>integral-source-1</th>
<th>integral-fission-1</th>
<th>integral-in-green-1</th>
<th>integral-self-react-1</th>
<th>integral-aborption-1</th>
<th>integral-right leakage-1</th>
<th>integral-net leakage-1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9.93401746-01</td>
<td>0.00000000-00</td>
<td>1.00000000-00</td>
<td>4.91572768-01</td>
<td>4.14684225-01</td>
<td>9.40787236-01</td>
<td>5.92151828-02</td>
<td>1.00000000-00</td>
</tr>
</tbody>
</table>

**execution time** 0.47 seconds

******************************************************************************

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

******************************************************************************
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard k-eff Calculation

...isotropic flux component...

<table>
<thead>
<tr>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.457023</td>
<td>2.398534</td>
<td>2.339456</td>
<td>2.280345</td>
<td>2.221234</td>
</tr>
<tr>
<td>2.398534</td>
<td>2.339456</td>
<td>2.280345</td>
<td>2.221234</td>
<td>2.162123</td>
</tr>
<tr>
<td>2.339456</td>
<td>2.280345</td>
<td>2.221234</td>
<td>2.162123</td>
<td>2.103030</td>
</tr>
<tr>
<td>2.280345</td>
<td>2.221234</td>
<td>2.162123</td>
<td>2.103030</td>
<td>2.043939</td>
</tr>
<tr>
<td>2.221234</td>
<td>2.162123</td>
<td>2.103030</td>
<td>2.043939</td>
<td>2.000000</td>
</tr>
</tbody>
</table>

...fission source rate...

<table>
<thead>
<tr>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.043939</td>
<td>2.043939</td>
<td>2.043939</td>
<td>2.043939</td>
<td>2.043939</td>
</tr>
<tr>
<td>2.000000</td>
<td>2.000000</td>
<td>2.000000</td>
<td>2.000000</td>
<td>2.000000</td>
</tr>
<tr>
<td>1.956753</td>
<td>1.956753</td>
<td>1.956753</td>
<td>1.956753</td>
<td>1.956753</td>
</tr>
<tr>
<td>1.910453</td>
<td>1.910453</td>
<td>1.910453</td>
<td>1.910453</td>
<td>1.910453</td>
</tr>
<tr>
<td>1.864153</td>
<td>1.864153</td>
<td>1.864153</td>
<td>1.864153</td>
<td>1.864153</td>
</tr>
</tbody>
</table>

...criticality calculation...

<table>
<thead>
<tr>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
<th>zone pt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.817852</td>
<td>1.817852</td>
<td>1.817852</td>
<td>1.817852</td>
<td>1.817852</td>
</tr>
<tr>
<td>1.771552</td>
<td>1.771552</td>
<td>1.771552</td>
<td>1.771552</td>
<td>1.771552</td>
</tr>
<tr>
<td>1.725252</td>
<td>1.725252</td>
<td>1.725252</td>
<td>1.725252</td>
<td>1.725252</td>
</tr>
<tr>
<td>1.678952</td>
<td>1.678952</td>
<td>1.678952</td>
<td>1.678952</td>
<td>1.678952</td>
</tr>
<tr>
<td>1.632652</td>
<td>1.632652</td>
<td>1.632652</td>
<td>1.632652</td>
<td>1.632652</td>
</tr>
</tbody>
</table>

...interface file created...

...interface file flux written...

ONEDANT USER'S GUIDE Version 3.0
Sample Problem 1: Standard kэфф Calculation

run highlights

* all modules are tentatively go.
* interface file produced, written.
  cross sections from cards.
* interface mixing file written.
* interface file aggregate written.
* xs files main, output written.
* left boundary condition overridden.
* interface file soluth file written.
  edit module execution suppressed.
  neither editit nor edit cards exist.
  start silver execution.
  all convergence criteria met.
  interface file ntimes written.
  interface file nfiles written.

storage and timing history

<table>
<thead>
<tr>
<th>module</th>
<th>sec</th>
<th>words</th>
<th>limit</th>
<th>sec</th>
<th>limit</th>
<th>cpu</th>
<th>avg cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.0</td>
<td>0.0</td>
</tr>
<tr>
<td>100</td>
<td>3706</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.2</td>
<td>0.0</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>0.0</td>
</tr>
<tr>
<td>102</td>
<td>1038</td>
<td>40000</td>
<td>5</td>
<td>14000</td>
<td>0.1</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>92</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>2074</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>215</td>
<td>40000</td>
<td>92</td>
<td>14000</td>
<td>0.1</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>470</td>
<td>40000</td>
<td>7</td>
<td>14000</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>1334</td>
<td>40000</td>
<td>670</td>
<td>14000</td>
<td>0.8</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>201</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>202</td>
<td>261</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>203</td>
<td>261</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>204</td>
<td>1334</td>
<td>40000</td>
<td>39</td>
<td>14000</td>
<td>0.1</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>205</td>
<td>1299</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>206</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>207</td>
<td>1310</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>208</td>
<td>152</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>209</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
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 $\Sigma_f$ and $\Sigma_c$ 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.
Appendix A: Sample Input

Sample Problem 2: Edit-Only Run

...edit output...

...block vi - edit specification data...

**Key start edit output.*

**cross-section balancing (subcoo, etc.)**

or

**transport correction (transport, cesium, or bns)**

will NOT be reflected in edits

...input control integers...

1 iout 0/1 no/yes - print edit status

1 zout 0/1 no/yes - zone edits desired

0 ajed 0/1 direct/adjoint edit (use rflux/rflux file)

0 zipped 0/1/2/3 print totals only/print broad groups only/issue as 1/print all groups and totals

0 byvolp 0/1 no/yes - multiply polar reaction rates by mesh volumes

0 rflux 0/1 no/yes - write the rflux file (zone average flux file)

0 remfix 0/1 no/yes - write the remfix file (zone average flux moments file)

...floating parameters...

0.0000000000 power 0/p no/normalize all results, including flux files, to p regevatts

2.1000000000 mepscr 0/m 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

(required atomic weights to be present)

...edit output...

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

...description of cross section edits...

<table>
<thead>
<tr>
<th>isotope no.</th>
<th>material no.</th>
<th>constituent no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>fuel</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>resid</td>
<td>u-235</td>
</tr>
</tbody>
</table>

reaction rates will be framed for each of the above

using the cross-section types shown below

<table>
<thead>
<tr>
<th>position</th>
<th>type no.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>matgrf</td>
</tr>
<tr>
<td></td>
<td>files</td>
</tr>
</tbody>
</table>

Onedant User's Guide  Version 3.0  2-89
**APPENDIX A: SAMPLE INPUT**

Sample Problem 2: Edit-Only Run

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

******************************************************************************

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

... permanent material 2 - fuel ...

<table>
<thead>
<tr>
<th>point</th>
<th>av</th>
<th>ruseig</th>
<th>fice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>8.122316E-04</td>
<td>2.65651E-04</td>
</tr>
<tr>
<td>2</td>
<td>3.000</td>
<td>8.094971E-04</td>
<td>2.62304E-04</td>
</tr>
<tr>
<td>3</td>
<td>5.000</td>
<td>8.013905E-04</td>
<td>2.63234E-04</td>
</tr>
<tr>
<td>4</td>
<td>7.000</td>
<td>7.941938E-04</td>
<td>2.62092E-04</td>
</tr>
<tr>
<td>5</td>
<td>9.000</td>
<td>7.820006E-04</td>
<td>2.59988E-04</td>
</tr>
<tr>
<td>6</td>
<td>11.000</td>
<td>7.677782E-04</td>
<td>2.51614E-04</td>
</tr>
<tr>
<td>7</td>
<td>13.000</td>
<td>7.505778E-04</td>
<td>2.53477E-04</td>
</tr>
<tr>
<td>8</td>
<td>15.000</td>
<td>7.330952E-04</td>
<td>2.53938E-04</td>
</tr>
<tr>
<td>9</td>
<td>17.000</td>
<td>7.081432E-04</td>
<td>2.52102E-04</td>
</tr>
<tr>
<td>10</td>
<td>19.000</td>
<td>6.854624E-04</td>
<td>2.51075E-04</td>
</tr>
<tr>
<td>46</td>
<td>51.000</td>
<td>1.627576E-05</td>
<td>5.62631E-06</td>
</tr>
<tr>
<td>47</td>
<td>51.000</td>
<td>1.406848E-05</td>
<td>4.68189E-06</td>
</tr>
<tr>
<td>48</td>
<td>51.000</td>
<td>1.292225E-05</td>
<td>3.79622E-06</td>
</tr>
<tr>
<td>49</td>
<td>51.000</td>
<td>8.483782E-06</td>
<td>2.81997E-06</td>
</tr>
<tr>
<td>50</td>
<td>51.000</td>
<td>5.952696E-06</td>
<td>1.84478E-06</td>
</tr>
</tbody>
</table>

... resident macroscopic ...

<table>
<thead>
<tr>
<th>point</th>
<th>av</th>
<th>ruseig</th>
<th>fice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000</td>
<td>2.840292E-04</td>
<td>9.31568E-05</td>
</tr>
<tr>
<td>2</td>
<td>3.000</td>
<td>2.815342E-04</td>
<td>9.28139E-05</td>
</tr>
<tr>
<td>3</td>
<td>5.000</td>
<td>2.811832E-04</td>
<td>9.21396E-05</td>
</tr>
<tr>
<td>4</td>
<td>7.000</td>
<td>2.782275E-04</td>
<td>9.10971E-05</td>
</tr>
<tr>
<td>5</td>
<td>9.000</td>
<td>2.758754E-04</td>
<td>9.07933E-05</td>
</tr>
<tr>
<td>6</td>
<td>11.000</td>
<td>2.667576E-04</td>
<td>9.05618E-05</td>
</tr>
<tr>
<td>7</td>
<td>13.000</td>
<td>2.615720E-04</td>
<td>9.00814E-05</td>
</tr>
<tr>
<td>8</td>
<td>15.000</td>
<td>2.594715E-04</td>
<td>8.98036E-05</td>
</tr>
<tr>
<td>9</td>
<td>17.000</td>
<td>2.479132E-04</td>
<td>8.87387E-05</td>
</tr>
<tr>
<td>10</td>
<td>19.000</td>
<td>2.392758E-04</td>
<td>8.79489E-05</td>
</tr>
<tr>
<td>46</td>
<td>51.000</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>47</td>
<td>51.000</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>48</td>
<td>51.000</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>49</td>
<td>51.000</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>50</td>
<td>51.000</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
</tbody>
</table>

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

... permanent material 2 - fuel ...

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>ruseig</th>
<th>fice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.02569E-03</td>
<td>2.76765E-06</td>
<td>9.07858E-01</td>
</tr>
<tr>
<td>2</td>
<td>1.03367E-04</td>
<td>1.34892E-06</td>
<td>4.16544E-01</td>
</tr>
<tr>
<td>3</td>
<td>1.02269E-04</td>
<td>4.04091E-06</td>
<td>1.31b62E-01</td>
</tr>
<tr>
<td>sum</td>
<td>3.1416E-04</td>
<td>4.55917E-06</td>
<td>1.50083E-00</td>
</tr>
</tbody>
</table>

... resident macroscopic ...

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>ruseig</th>
<th>fice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.02569E-03</td>
<td>9.69256E-06</td>
<td>3.17751E-01</td>
</tr>
<tr>
<td>2</td>
<td>1.03367E-04</td>
<td>2.43452E-06</td>
<td>8.13698E-03</td>
</tr>
<tr>
<td>3</td>
<td>1.02269E-04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>sum</td>
<td>3.1416E-04</td>
<td>9.34402E-06</td>
<td>1.25876E-01</td>
</tr>
</tbody>
</table>
******************************************************************************

2-90 ONEDANT USER'S GUIDE Version 3.0
**APPENDIX A: SAMPLE INPUT**

Sample Problem 2: Edit-Only Run

---

```
*Key start constituents*

**reaction rates**
from constituents

- point edit for the sum of the neutron groups...

- constituent isotope 6 - Pu-239...

<table>
<thead>
<tr>
<th>point</th>
<th>av rnd</th>
<th>nusegf</th>
<th>files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000</td>
<td>2.6108E-04</td>
<td>8.5413E-05</td>
</tr>
<tr>
<td>2</td>
<td>3.0000</td>
<td>2.6219E-04</td>
<td>8.5148E-05</td>
</tr>
<tr>
<td>3</td>
<td>5.0000</td>
<td>2.5821E-04</td>
<td>8.4743E-05</td>
</tr>
<tr>
<td>4</td>
<td>7.0000</td>
<td>2.5536E-04</td>
<td>8.3331E-05</td>
</tr>
<tr>
<td>5</td>
<td>9.0000</td>
<td>2.5153E-04</td>
<td>8.2291E-05</td>
</tr>
<tr>
<td>6</td>
<td>11.0000</td>
<td>2.4947E-04</td>
<td>8.0760E-05</td>
</tr>
<tr>
<td>7</td>
<td>13.0000</td>
<td>2.4715E-04</td>
<td>7.9597E-05</td>
</tr>
<tr>
<td>8</td>
<td>15.0000</td>
<td>2.4494E-04</td>
<td>7.8717E-05</td>
</tr>
<tr>
<td>9</td>
<td>17.0000</td>
<td>2.4279E-04</td>
<td>7.7349E-05</td>
</tr>
<tr>
<td>10</td>
<td>19.0000</td>
<td>2.3990E-04</td>
<td>7.5041E-05</td>
</tr>
<tr>
<td>46</td>
<td>91.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>47</td>
<td>93.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>48</td>
<td>95.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>49</td>
<td>97.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>50</td>
<td>99.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

- constituent isotope 7 - U-238...

<table>
<thead>
<tr>
<th>point</th>
<th>av rnd</th>
<th>nusegf</th>
<th>files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000</td>
<td>2.3259E-05</td>
<td>7.7419E-06</td>
</tr>
<tr>
<td>2</td>
<td>3.0000</td>
<td>2.3154E-05</td>
<td>7.1163E-06</td>
</tr>
<tr>
<td>3</td>
<td>5.0000</td>
<td>2.2973E-05</td>
<td>6.6578E-06</td>
</tr>
<tr>
<td>4</td>
<td>7.0000</td>
<td>2.2701E-05</td>
<td>6.2457E-06</td>
</tr>
<tr>
<td>5</td>
<td>9.0000</td>
<td>2.2478E-05</td>
<td>5.8727E-06</td>
</tr>
<tr>
<td>6</td>
<td>11.0000</td>
<td>2.2190E-05</td>
<td>5.5339E-06</td>
</tr>
<tr>
<td>7</td>
<td>13.0000</td>
<td>2.1301E-05</td>
<td>5.1935E-06</td>
</tr>
<tr>
<td>8</td>
<td>15.0000</td>
<td>2.0794E-05</td>
<td>4.8886E-06</td>
</tr>
<tr>
<td>9</td>
<td>17.0000</td>
<td>2.0212E-05</td>
<td>4.6194E-06</td>
</tr>
<tr>
<td>10</td>
<td>19.0000</td>
<td>1.9743E-05</td>
<td>4.3829E-06</td>
</tr>
<tr>
<td>46</td>
<td>91.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>47</td>
<td>93.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>48</td>
<td>95.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>49</td>
<td>97.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>50</td>
<td>99.0000</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
</tbody>
</table>

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

- constituent isotope 6 - Pu-239...

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>nusegf</th>
<th>files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.0285E+03</td>
<td>8.9232E-01</td>
<td>2.8947E-01</td>
</tr>
<tr>
<td>2</td>
<td>1.0367E+04</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>3</td>
<td>1.8023E+04</td>
<td>2.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>3.1415E+04</td>
<td>8.9323E-01</td>
<td>2.8947E-01</td>
<td></td>
</tr>
</tbody>
</table>

- constituent isotope 7 - U-238...

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>nusegf</th>
<th>files</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.0285E+03</td>
<td>7.5023E-02</td>
<td>2.5274E-02</td>
</tr>
<tr>
<td>2</td>
<td>1.6386E+04</td>
<td>2.4345E-02</td>
<td>8.1163E-03</td>
</tr>
<tr>
<td>3.1415E+04</td>
<td>1.0017E+01</td>
<td>3.3390E-02</td>
<td></td>
</tr>
</tbody>
</table>
```

---

**ONEDANT USER'S GUIDE** Version 3.0 2-91
### Sample Problem 2: Edit-Only Run

#### Run Highlights

- Solver module execution suppressed.
- Interface file edit: write.
- Start edit expansion.
- Edits completed.

#### Storage and Timing History

<table>
<thead>
<tr>
<th>Module</th>
<th>ECM</th>
<th>ACM</th>
<th>ECM Limit</th>
<th>ACM Limit</th>
<th>CPU</th>
<th>Sysct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.3</td>
<td>0.0</td>
</tr>
<tr>
<td>100</td>
<td>3706</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.6</td>
<td>0.0</td>
</tr>
<tr>
<td>102</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>103</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>108</td>
<td>3722</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>0.0</td>
</tr>
<tr>
<td>110</td>
<td>1308</td>
<td>40000</td>
<td>289</td>
<td>140000</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
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:

\[ \text{dant.x} \quad < \quad \text{odninp} \quad > \quad \text{odnout} \]

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>
<thead>
<tr>
<th>LIB</th>
<th>SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>Current Working Directory (CWD).</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>SNXSPATH, then CWD, but see text below.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>None, the library is contained in the input file.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>CWD</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>CWD</td>
</tr>
<tr>
<td>MENDFA</td>
<td>Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.</td>
</tr>
<tr>
<td>MENDFGB</td>
<td></td>
</tr>
<tr>
<td>XSLIB</td>
<td>SNXSPATH, then CWD</td>
</tr>
<tr>
<td>other</td>
<td>For any name other than those above, the program will assume the form is XSLIB and search for it in SNXS-PATH, then CWD.</td>
</tr>
</tbody>
</table>

---

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
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

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

This work was supported by the US Department of Energy.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
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

**TABLE OF CONTENTS** ........................................................................................................ 3-5

**LIST OF FIGURES** ........................................................................................................ 3-7

**LIST OF TABLES** .......................................................................................................... 3-9

**INTRODUCTION** ............................................................................................................ 3-11

**DOCUMENTATION FOR TWODANT USAGE** .............................................................. 3-13

- What Is In This User's Guide .......................................................................................... 3-13
- What Is Available Elsewhere ........................................................................................... 3-14

**TWODANT INPUT OVERVIEW** .................................................................................. 3-17

- Input Block Order ......................................................................................................... 3-17
- Free Field Input Summary .............................................................................................. 3-19
  - Arrays .......................................................................................................................... 3-19
  - Numeric Data Items .................................................................................................... 3-19
  - Character Data Items .................................................................................................. 3-19
  - Blocks .......................................................................................................................... 3-20
  - Strings .......................................................................................................................... 3-20
  - Comments ...................................................................................................................... 3-20
  - Operators ....................................................................................................................... 3-20
  - Frequently Used Operators ......................................................................................... 3-21
- MINI-MANUAL Introduction .......................................................................................... 3-22
- MINI MANUAL .............................................................................................................. 3-23

**TWODANT INPUT DETAILS** ....................................................................................... 3-29

- Introduction ................................................................................................................... 3-29
- Title Line Details .......................................................................................................... 3-32
  - Title Line Control ...................................................................................................... 3-32
- Block-I Details: Dimensions and Controls ..................................................................... 3-33
  - Dimensions ............................................................................................................... 3-33
  - Storage Requirements ............................................................................................... 3-33
  - Minimum Print Output ............................................................................................... 3-34
  - Run Configuration Controls .................................................................................... 3-35
- Block-II Details: Geometry .......................................................................................... 3-36
  - Geometry Arrays ........................................................................................................ 3-36
- Block-III Details: Nuclear Data ...................................................................................... 3-37
  - Nuclear Data Type and Options ............................................................................... 3-37
  - Alternate Library Name ............................................................................................. 3-39
  - Text Cross-Section Library Format ......................................................................... 3-41
- Block-IV Details: Cross-Section Mixing ....................................................................... 3-43
  - MAILS input array ..................................................................................................... 3-44
  - Primary Mixing Arrays .............................................................................................. 3-44
  - ASSIGN input array .................................................................................................. 3-45
  - PREMIX input array .................................................................................................. 3-45
  - Character Names vs. Numeric Names ....................................................................... 3-46
  - Mixing Array for a Concentration Search .................................................................. 3-46
  - ASGMOD input array .................................................................................................. 3-47
  - Concentration Modifier ............................................................................................. 3-47
  - Fine Mesh Mixing ...................................................................................................... 3-48
TABLE OF CONTENTS

Miscellaneous Mixing Input ................................................................. 3-49
Block-V Details: Solver Input ................................................................. 3-50
Desired Calculation .............................................................................. 3-50
Iteration Controls ................................................................................ 3-51
Acceleration Controls .......................................................................... 3-51
K-Code Convergence ........................................................................... 3-51
Output Controls .................................................................................... 3-52
Miscellaneous Solver Input ................................................................. 3-53
Quadrature Details ................................................................................ 3-54
Flux Guess From a File .......................................................................... 3-55
General Eigenvalue Search Control ...................................................... 3-55
Dimension Search Input ........................................................................ 3-56
Concentration Search Input .................................................................. 3-56
Volumetric Source Options ................................................................. 3-57
Boundary Source Input ......................................................................... 3-59
First Collision Source Input ................................................................. 3-60
Monte Carlo Option Controls ............................................................... 3-61
Monte Carlo Biasing Input ..................................................................... 3-62
Monte Carlo Source Input ..................................................................... 3-63

Block-VI Details: Edit Input ................................................................... 3-65
Edit Spatial Specifications ...................................................................... 3-65
Reaction Rates from Cross Sections ...................................................... 3-66
Edit Cross-Section Types by Position and Name ................................... 3-67
Reaction Rates from User Response Functions .................................... 3-68
Energy Group Collapse Specifications .................................................. 3-69
Reaction Rate Summing ........................................................................ 3-70
Mass Inventories .................................................................................... 3-70
Power Normalization ............................................................................ 3-71
Miscellaneous Edit Items ...................................................................... 3-72
Special Plot Linkage ............................................................................. 3-73
MENDF Library Edit Cross Sections .................................................... 3-74

REFERENCES ......................................................................................... 3-75

APPENDIX A: SAMPLE INPUT ............................................................... 3-77
Sample Problem 1: Standard $k_{eff}$ Calculation .................................... 3-77
Sample Problem 1: Output Description .................................................. 3-77
Sample Problem 1: Output Listing ......................................................... 3-79
Sample Problem 2: Coupled $S_n$ Monte Carlo Calculation .................... 3-97
Sample Problem 2: Output Description .................................................. 3-97
Sample Problem 2: Output Listing ......................................................... 3-102

APPENDIX B: OPERATING SYSTEM SPECIFICS .................................. 3-123
UNIX/UNICOS Execution ...................................................................... 3-123
Library Search Path .............................................................................. 3-124
LIST OF FIGURES

Figure 3.1: TWODANT Input Order ................................................................. 3-18
LIST OF TABLES

Table 3.1: LIBNAME Availability ......................................................... 3-39
Table 3.2: UNIX Search Path ................................................................. 3-124
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.\textsuperscript{1,2} It is a two-dimensional version of the one-dimensional code, ONEDANT.\textsuperscript{3}

TWODANT\textsuperscript{TM} is based on the modular construction of the DANTSYS\textsuperscript{TM} 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\textsuperscript{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\textsuperscript{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,\textsuperscript{6}

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

12. a Monte Carlo/discrete-ordinates (MC/S\textsubscript{n}) option for solving all or part of the problem geometry using Monte Carlo,\textsuperscript{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.
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.
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
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:

\[ \text{SOURCX}=0\ 0\ 0\ 2\ R\ 1.1\ F0 \]

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 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

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

\(^a\) 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
(316 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

**Block-IV: Mixing**

MATLS [:-:MT]
ASSIGN [:-:NZONE]
PREMIX [:-:]
ASGMOD [:-:]
CMOD
FMMIX

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

ATWT [:]

T

**Block-V: Solver**

IEVT
ISCT
ITH
IBL
IBR
IBT
IBB

EPSI
IITL
IITM
OITM
ITLIM
NOSIGF

KCALC

iff LIB=ODNINP, insert ASCII text cross sections here
### Solver (continued)

--- Output Controls ---

- FLUXP
- XSECTP
- FISSLRP
- 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 -
**SOURCE [NGROUP;NMQ] and 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]**

--- Monte Carlo Option ---

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

--- Monte Carlo Biasing ---

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

--- Monte Carlo Sources ---

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

T

### First Collision Source ---

**FCSRC**
valid:
**ANA**
**RAY**
**NO**

**FCNRAY**
**FCNTR**
**FCWCO**

--- Monte Carlo Biasing ---

**MCSB**
**MCWC [2]**
**MCMMIMP [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; -]
RSFNAME [-]

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}

<table>
<thead>
<tr>
<th>Word</th>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHEAD</td>
<td>Number of title lines that follow.(^b)</td>
</tr>
<tr>
<td>2</td>
<td>NOTTY</td>
<td>Suppress output to on-line user terminal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0/1 = no/yes.</td>
</tr>
<tr>
<td>3</td>
<td>NOLIST</td>
<td>Suppress listing of all ASCII text input?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0/1 = no/yes. (default=no)</td>
</tr>
</tbody>
</table>

\(^a\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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   | $S_p$ 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. |

---

Storage Requirements

{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSCM</td>
<td>Length of SCM desired (default=$40000_{10}$)</td>
</tr>
<tr>
<td>MAXLCM</td>
<td>Length of LCM desired (default=$140000_{10}$)</td>
</tr>
</tbody>
</table>

---

\(^a\) 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.
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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINIPRT</td>
<td>Provide a shortened output print file. a 0/1 = no/yes b</td>
</tr>
</tbody>
</table>

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\}

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

Note: Default on all these controls is no.
## Block-II Details: Geometry

### Geometry Arrays\(^a\)  
\{Required\}

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

---

\(^a\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIB</strong></td>
<td>Name and form of the cross-section data file. Enter as a data item one of the following words:</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>ISOTXS&lt;sup&gt;b&lt;/sup&gt;</td>
<td>CCCC standard isotope ordered binary cross-section file.</td>
</tr>
<tr>
<td>XSLIB</td>
<td>ASCII text library supplied in a separate file named XSLIB.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>ASCII text library follows after this block of input (after the T of Block-III).</td>
</tr>
<tr>
<td>GRUPXS&lt;sup&gt;c&lt;/sup&gt;</td>
<td>CCCC standard group ordered cross-section file.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Binary library supplied as a separate file named BXSLIB. (See “Binary Form of Card-Image Libraries (the BXSLIB file)” on page 10-12.</td>
</tr>
<tr>
<td>MACRXS&lt;sup&gt;d&lt;/sup&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>See “XSLIBB Card-Image Library File” on page 10-12.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>ASCII form of MACRXS file.</td>
</tr>
<tr>
<td>MENDF</td>
<td>(LANL only) See “The Los Alamos MENDF5 Cross-Section Library” on page 10-13.</td>
</tr>
<tr>
<td>MENDFG</td>
<td>(LANL only) See “The Los Alamos MENDF5G Gamma Cross-Section Library” on page 10-14.</td>
</tr>
<tr>
<td>other</td>
<td>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.</td>
</tr>
<tr>
<td>WRITMXS&lt;sup&gt;e&lt;/sup&gt;</td>
<td>Controls the code’s writing certain ASCII cross-section files. Enter one of the following words:</td>
</tr>
</tbody>
</table>

---

<sup>a</sup> Enter as a data item one of the following words: ISOTXS, XSLIB, ODNINP, GRUPXS, BXSLIB, MACRXS, XSLIBB, MACBCD, MENDF, MENDFG, other.

<sup>b</sup> CCCC standard isotope ordered binary cross-section file.

<sup>c</sup> CCCC standard group ordered cross-section file.

<sup>d</sup> Binary library supplied as a separate file named BXSLIB. (See “Binary Form of Card-Image Libraries (the BXSLIB file)” on page 10-12.

<sup>e</sup> ASCII cross-section files.
### Nuclear Data Type and Options (Cont.)

#### Required

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>MACBCD</td>
<td>Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file.</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file.</td>
</tr>
<tr>
<td>XSLIBE</td>
<td>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).</td>
</tr>
<tr>
<td>XSLIBF</td>
<td>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).</td>
</tr>
<tr>
<td>LNG</td>
<td>Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.</td>
</tr>
<tr>
<td>BALXS</td>
<td>Cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using!</td>
</tr>
<tr>
<td>NTICHI</td>
<td>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.</td>
</tr>
<tr>
<td>CHIVEC</td>
<td>Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI in Block-V.</td>
</tr>
<tr>
<td>[NGROUP]</td>
<td>{optional} Cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using!</td>
</tr>
</tbody>
</table>

*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 12LP1=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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBNAME</td>
<td>Alternate name of the library file. May be used only with certain types of libraries. See Table 3.1.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>LIB</th>
<th>LIBNAME AVAILABLE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>No</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>Yes</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>Yes</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>ODNINP</td>
<td>No</td>
</tr>
<tr>
<td>MACBCD</td>
<td>No</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>No</td>
</tr>
<tr>
<td>MENDF(^a)</td>
<td>No</td>
</tr>
<tr>
<td>MENDFG(^b)</td>
<td>No</td>
</tr>
<tr>
<td>XSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>other</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXORD</td>
<td>Highest Legendre order in the scattering tables.</td>
</tr>
<tr>
<td>IHM</td>
<td>Number of positions (entries) in each row of the cross-section table.</td>
</tr>
<tr>
<td>IHT</td>
<td>Position number of the total cross section.</td>
</tr>
<tr>
<td>IHS</td>
<td>Position number of the self-scatter cross section. (default = IHT + 1).</td>
</tr>
<tr>
<td>IFIDO</td>
<td>Format of the cross-section library. (-1/0/1/2 = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field).</td>
</tr>
<tr>
<td>ITITL</td>
<td>A title line precedes each table. 0/1 = no/yes</td>
</tr>
<tr>
<td>I2LPL1</td>
<td>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.</td>
</tr>
<tr>
<td>SAVBXS</td>
<td>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.</td>
</tr>
<tr>
<td>KWIKRD</td>
<td>Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? 0/1 = no/yes (default=yes).</td>
</tr>
<tr>
<td>NAMES [NISO]</td>
<td>Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).</td>
</tr>
<tr>
<td>EDNAME [IHT-3]</td>
<td>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.).</td>
</tr>
<tr>
<td>NTPI [NISO]</td>
<td>Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).</td>
</tr>
<tr>
<td>VEL [NGROUP]</td>
<td>Speeds for each group. Needed only for alpha calculations.</td>
</tr>
<tr>
<td>EBOUND [NGROUP+1]</td>
<td>Energy boundaries for each group.</td>
</tr>
</tbody>
</table>
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:

\[ \ldots \sigma_{abs}, \sigma_t, \sigma_{total}, \ldots \sigma_{g+2 \rightarrow g}, \sigma_{g+1 \rightarrow g}, \sigma_{g \rightarrow g}, \sigma_{g-1 \rightarrow g}, \sigma_{g-2 \rightarrow g}, \ldots \]

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\textsuperscript{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.
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 \( \Sigma_g \) 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 \( \Sigma_g \) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLS(^a) [;-;MT]</td>
<td>Instructions for mixing &quot;isotopes&quot; or premixes into &quot;materials.&quot; See details below.</td>
</tr>
<tr>
<td>ASSIGN(^b) [;-;NZONE]</td>
<td>Assignments of materials to geometric zones. See below.</td>
</tr>
<tr>
<td>PREMIX [;-;] {optional}</td>
<td>Instructions for mixing &quot;isotopes&quot; into premixes. See below.</td>
</tr>
</tbody>
</table>

\(^a\) 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:

\[ \text{MATLS}= mat_1 \; \text{comp}_1 \; \text{den}_1, \; \text{comp}_2 \; \text{den}_2, \; \text{etc.} \ldots ; \]

where \( mat_1 \) is the desired character name of the first material and \( \text{comp}_1, \text{comp}_2 \), and so on are the character names of its components which have "densities" of, respectively, \( \text{den}_1, \text{den}_2 \), 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 \( \text{den}_i \) 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:

\[
\text{MATLS}= \text{mat}_1 \text{ comp}_1; \quad \text{mat}_2 \text{ comp}_2 \text{ den}_2; \quad \ldots \text{etc.} \ldots ;
\]

**ASSIGN input array**

The general form of the ASSIGN instruction is shown below:

\[
\text{ASSIGN}= \text{zone}_1 \text{ mat}_1 \text{ vol}_1, \text{ mat}_2 \text{ vol}_2, \ldots \text{etc.} \ldots ;
\]

where \( \text{zone}_1 \) is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). \( \text{mat}_1, \text{mat}_2 \), and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" \( \text{vol}_1, \text{vol}_2 \), and so on. Additional zones (i.e., \( \text{zone}_2, \text{zone}_3 \), and so on up to the required number, \( \text{NZONE} \)) 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:

\[
\text{PREMIX}= \text{mat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \ldots \text{etc.} \ldots ;
\]
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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASGMOD(^a) ([-;]-)</td>
<td>( C_1 ) parameters used in concentration searches. See the discussion below.</td>
</tr>
</tbody>
</table>

\(^a\) 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:

\[ C(z,x) = C_0(z,x) + C_1(z,x) \times CMOD \]

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:

\[ \text{ASGMOD} = \text{zone mat}_m \text{ vol}_m, \text{ mat}_n \text{ vol}_n, \ldots \text{etc.} ; \]

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOD</td>
<td>Concentration modifier. Input value is not used in a search. See the discussion below.</td>
</tr>
</tbody>
</table>
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**

*Optional*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMMIX</td>
<td>Read the composition of each fine mesh from the file LNK3DNT. ( 0/1 = \text{no/yes} ).</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATNAM [MT]</td>
<td>Character material names for Materials. Used only if the \textit{mat}_i 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.</td>
</tr>
<tr>
<td>ZONNAM [NZONE]</td>
<td>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.</td>
</tr>
<tr>
<td>MATSPEC [\leq MT]</td>
<td>Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions.</td>
</tr>
<tr>
<td></td>
<td>Allowable entries are the words:</td>
</tr>
<tr>
<td></td>
<td>\textit{ATDENS} (default) atomic densities</td>
</tr>
<tr>
<td></td>
<td>\textit{ATFRAC}^{a} atomic fractions</td>
</tr>
<tr>
<td></td>
<td>\textit{WTFRAC} weight fractions</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>ATWT [\leq 2*NISO]</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>\text{ATWT} = iso_1 \text{atwt}_1 iso_2 \text{atwt}_2 ....</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>[See “Using Atomic Fractions or Weight Fractions (MATSPEC)” on page 11-13].</td>
</tr>
</tbody>
</table>

\text{a.} ATFRAC and WTFRAC cannot be used with PREMIX.

\text{b.} Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.
## Block-V Details: Solver Input

### Desired Calculation

<table>
<thead>
<tr>
<th>Name</th>
<th>Calculation type: Enter one of the following values:</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEVT</td>
<td>Value</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td>ISCT</td>
<td>Legendre order of scattering (default = 0).</td>
</tr>
<tr>
<td>ITH</td>
<td>$0/1$ = direct/adjoint calculation (default = 0).</td>
</tr>
<tr>
<td>IBL</td>
<td>Left boundary condition(^a). $0/1/2/3 = \text{vacuum/reflective/periodic/white}$ (default = vacuum).</td>
</tr>
<tr>
<td>IBR</td>
<td>Right boundary condition. $0/1/2/3 = \text{vacuum/reflective/periodic/white}$ (default = vacuum).</td>
</tr>
<tr>
<td>IBT</td>
<td>Top boundary condition. $0/1/2/3 = \text{vacuum/reflective/periodic/white}$ (default = vacuum).</td>
</tr>
<tr>
<td>IBB</td>
<td>Bottom boundary condition. $0/1/2/3 = \text{vacuum/reflective/periodic/white}$ (default = vacuum).</td>
</tr>
</tbody>
</table>

\(^a\) The left boundary condition applies only for x,y geometry.
Iteration Controls

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPSI</td>
<td>Convergence precision (default=0.0001).</td>
</tr>
<tr>
<td>IITL</td>
<td>Maximum number of inner iterations per group at first (default=1).</td>
</tr>
<tr>
<td>IITM</td>
<td>Maximum number of inner iterations allowed when fission source is near convergence (default chosen by code).</td>
</tr>
<tr>
<td>OITM</td>
<td>Maximum number of outer iterations (default=20).</td>
</tr>
<tr>
<td>ITLIM</td>
<td>Number of seconds time limit (default=unlimited).</td>
</tr>
<tr>
<td>NOSIGF&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Inhibit fission multiplication in a fixed source problem. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>

<sup>a</sup> 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 \(\Phi\) 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREYACC</td>
<td>Upscatter acceleration&lt;sup&gt;a&lt;/sup&gt;. 0/1 = no/yes&lt;sup&gt;b&lt;/sup&gt;.</td>
</tr>
</tbody>
</table>

<sup>a</sup> See page 7-21 for details.

<sup>b</sup> The user may also input the words no or yes.

K-Code Convergence

{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCALC</td>
<td>Special Criticality Convergence Scheme. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>
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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUXP</td>
<td>Final flux print. 0/1/2 = no/isotropic/all moments.</td>
</tr>
<tr>
<td>XSECTP</td>
<td>Cross-section print. 0/1/2 = no/principal/all.</td>
</tr>
<tr>
<td>FISSRP</td>
<td>Fission source rate print. 0/1 = no/yes.</td>
</tr>
<tr>
<td>SOURCP</td>
<td>Source print. 0/1/2/3 = no/as input/normalized/both.</td>
</tr>
<tr>
<td>ANGP</td>
<td>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.</td>
</tr>
<tr>
<td>BALP</td>
<td>Coarse Mesh Interval Print Options. Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>RAFLUX</td>
<td>Prepare angular flux file (RAFLUX or AAFLUX). 0/1 = no/yes.</td>
</tr>
<tr>
<td>RMFLUX</td>
<td>Prepare flux moments file (RMFLUX or AMFLUX). 0/1 = no/yes.</td>
</tr>
<tr>
<td>ASRITE</td>
<td>0/1 = No/Write right-going angular boundary flux from column i to file ARBFLUX. The value of i must be between 1 and IT+1.</td>
</tr>
<tr>
<td>ASBOTT</td>
<td>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.</td>
</tr>
<tr>
<td>ASTOP</td>
<td>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.</td>
</tr>
<tr>
<td>ASLEFT</td>
<td>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.</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRCOR</td>
<td>Apply transport correction to cross sections on MACRXS file. Enter one of the following words:</td>
</tr>
<tr>
<td></td>
<td><strong>Word</strong></td>
</tr>
<tr>
<td></td>
<td><strong>DIAG</strong></td>
</tr>
<tr>
<td></td>
<td><strong>BHS</strong></td>
</tr>
<tr>
<td></td>
<td><strong>CESARO</strong></td>
</tr>
<tr>
<td></td>
<td><strong>NO</strong></td>
</tr>
</tbody>
</table>

| 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 keff problems where the integral is equal to NORM*keff). 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. 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] | Density factor to use at each fine mesh point. Applied to the zone macroscopic cross sections at each mesh interval. |

| DENX [IT] | 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) for each group. 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 $\text{DEN}(i,j)$, at mesh interval $(i,j)$ is computed as follows:

$$\text{DEN}(i,j) = \text{DENX}(i) \times \text{DENY}(j)$$

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

### Quadrature Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRPSN [NGROUP]</td>
<td>$S_n$ order to be used for each group.</td>
</tr>
<tr>
<td>IQUAD</td>
<td>Source of quadrature constants. Enter one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>WGT$^a$ [MM]$^b$</td>
<td>Quadrature weights.</td>
</tr>
<tr>
<td>MU [MM]</td>
<td>Mu cosines.</td>
</tr>
<tr>
<td>ETA [MM]</td>
<td>Eta cosines.</td>
</tr>
</tbody>
</table>

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

b. $\text{MM} = \text{ISN} \times (\text{ISN}+2)/8$ for IQUAD = -2,1.

$\text{MM} = (\text{ISN}/2)**2$ for IQUAD = 2.
Flux Guess From a File
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFLUX</td>
<td>Read the initial flux guess from a file.(^a) 0/1 = no/yes. If ITH=0 and ISCT&gt;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&gt;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.</td>
</tr>
</tbody>
</table>

\(^a\) There is presently no text input flux guess available for TWODANT.

General Eigenvalue Search Control\(^a\)
{ IEVT >1 }

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

\(^a\) 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}*

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XM [IM]</td>
<td>x-dimension fractional change per coarse mesh (or r).</td>
</tr>
<tr>
<td>YM [JM]</td>
<td>y-dimension fractional change per coarse mesh (or z) (or theta).</td>
</tr>
</tbody>
</table>

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:

\[
XMSh_{i+1} = XMSh_i + \{XMSh_{i+1} - XMSh_i\} \times [1.0 + EV \times XM]_i, \quad i=1, \ldots, IM
\]

\[
YMesh_{j+1} = YMesh_j + \{YMesh_{j+1} - YMesh_j\} \times [1.0 + EV \times YM]_j, \quad j=1, \ldots, JM
\]

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}*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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 array in Block-IV.</td>
</tr>
</tbody>
</table>

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}

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

\(^a\) 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:

$$SOURCF(i,j,g,m) = SOURCE(g,m) \times SOURCX(i,m) \times SOURCY(j,m)$$
# Boundary Source Input

## {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SILEFT [NGROUP;JT]</td>
<td>Isotropic source on the left side. (Spectrum at each y mesh interval) (or z) (or theta).</td>
</tr>
<tr>
<td>SIRITE [NGROUP;JT]</td>
<td>Isotropic source on the right side.</td>
</tr>
<tr>
<td>SIBOTT [NGROUP;IT]</td>
<td>Isotropic source on the bottom side.</td>
</tr>
<tr>
<td>SITOP [NGROUP;IT]</td>
<td>Isotropic source on the top side.</td>
</tr>
</tbody>
</table>

**Option 1: Isotropic Boundary Source.**

**Option 2: Full Angular Boundary Source.**

**Option 3: Boundary Source From Vectors.**

---

### Notes:

- The order of the angles is identical to that used in the $S_\alpha$ 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.

- 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):

\[
S(m,g,j) = \text{BSLFTG}(g) * \text{BSLFTY}(j) * \text{BSLFTA}(m)
\]

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

---

### First Collision Source Input

**{Optional}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCSRC</td>
<td>Use First Collision Source Option. Enter one of the following words:</td>
</tr>
<tr>
<td>Word</td>
<td>Description</td>
</tr>
<tr>
<td>ANA(^a)</td>
<td>Use analytic first collision source.(^{12}) Cylindrical ((r,z)) geometry only.</td>
</tr>
<tr>
<td>RAY(^b)</td>
<td>Use the ray tracing first collision source, ((x,y)) or ((r,z)) geometry only.</td>
</tr>
<tr>
<td>NO</td>
<td>(or omit entry) - don't use</td>
</tr>
<tr>
<td>FCNRAY</td>
<td>Number of ray tracings/batch to use with ray tracing first collision source. (default = 10000).</td>
</tr>
<tr>
<td>FCNTR</td>
<td>Number of batches (trials) to use with ray tracing first collision source. If FCNTR is negative, read the restart file UCFLUX, and perform</td>
</tr>
<tr>
<td>FCWCO</td>
<td>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 ( \text{wtmin} * \text{FCWCO} ), where ( \text{wtmin} ) is the minimum source starting weight, then that ray is terminated. Default entry is 0.0, which corresponds to a FCWCO value of ( \text{EPSI}^{**2} ).</td>
</tr>
</tbody>
</table>

---

\(^a\) 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 \((1,j)\) mesh cell, where \( j \) 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:

\[
\text{SOURCY} = 14R0 1 FO
\]

- 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

{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCOPT</td>
<td>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).</td>
</tr>
<tr>
<td>MCREG [NGROUP]</td>
<td>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).</td>
</tr>
<tr>
<td>MCIBND [2]</td>
<td>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.</td>
</tr>
<tr>
<td>MCJBND [2]</td>
<td>Locates the bottom and top cells of the designated Monte Carlo region as described above. (default=0 0).</td>
</tr>
<tr>
<td>MCBLT</td>
<td>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).</td>
</tr>
<tr>
<td>MCITS</td>
<td>The maximum number of super-outer iterations allowed on the common boundary fluxes/sources between the Monte Carlo and S(_n) regions. (default=8).</td>
</tr>
<tr>
<td>MCNHSIS</td>
<td>The approximate number of histories per trial to be used in sampling the Monte Carlo source. (default=10000).</td>
</tr>
<tr>
<td>MCNTR</td>
<td>The number of trials (batches) to be used in sampling the Monte Carlo source. (default=25).</td>
</tr>
<tr>
<td>MCISEED</td>
<td>Resets the initial random seed. (default=0).</td>
</tr>
<tr>
<td>MCIPRNT</td>
<td>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.</td>
</tr>
</tbody>
</table>

\(a.\) 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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCSB</td>
<td>Biasing parameter for distributed source. Biases angular distribution of source according to $q(u)=C \times \exp(MCSB \times u)$, where $u$ is the direction cosine along the y (or z) axis.</td>
</tr>
<tr>
<td>MCWC [2]</td>
<td>Weight cutoff parameters $w_1$ and $w_2$. 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 $w_1 \times w_s$. $w_2$ must be less than or equal to $w_1$.</td>
</tr>
<tr>
<td>MCCMIMP [IM;JM]</td>
<td>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 $R_{ij} - \text{Int} R_{ij}$, $\text{Int} R_{ij} + 1$ particles are created; otherwise, $\text{Int} R_{ij}$ 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.</td>
</tr>
<tr>
<td>MCEGIMP [NGROUP]</td>
<td>Array of energy group importances for splitting/Russian roulette. Splitting/Russian roulette works as described above for coarse mesh cell importances.</td>
</tr>
<tr>
<td>MCLVIMP [ISN/2]</td>
<td>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.</td>
</tr>
</tbody>
</table>

---

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\(^a\)**}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCPTSRC [1/2]</td>
<td>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.</td>
</tr>
<tr>
<td>MCBSLA [2]</td>
<td>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.</td>
</tr>
<tr>
<td>MCBSRA [2]</td>
<td>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.</td>
</tr>
<tr>
<td>MCBSBA [2]</td>
<td>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.</td>
</tr>
<tr>
<td>MCBSTA [2]</td>
<td>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.</td>
</tr>
</tbody>
</table>

\(^a\) 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\(^a\)}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTED</td>
<td>Do edits by fine mesh. (0/1 = \text{no/yes}).</td>
</tr>
<tr>
<td>ZNED</td>
<td>Do edits by zone. (0/1 = \text{no/yes}). (i.e., edit zone, not SOLVER zone. See EDZONE input below).</td>
</tr>
<tr>
<td>POINTS[&lt;IT,JT&gt;]</td>
<td>Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points).</td>
</tr>
<tr>
<td></td>
<td>{optional}</td>
</tr>
<tr>
<td>EDZONE [IT,JT]</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>{optional}</td>
</tr>
</tbody>
</table>

\(^a\) 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\textsuperscript{a}
\{Optional\textsuperscript{b}\}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDXS [\leq\text{NEDT}] {required\textsuperscript{c}}</td>
<td>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 &quot;Edit Cross-Section Types by Position and Name&quot; on page 3-67 or &quot;MENDF Library Edit Cross Sections&quot; 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.</td>
</tr>
<tr>
<td>RESDNT</td>
<td>Do edits using the resident macroscopic cross section at each point. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDISOS [\leq\text{NISO}]</td>
<td>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).</td>
</tr>
<tr>
<td>EDCONS [\leq\text{NISO}]</td>
<td>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).</td>
</tr>
<tr>
<td>EDMATS [\leq\text{MT}]</td>
<td>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).</td>
</tr>
<tr>
<td>XDF\textsuperscript{d} [IT]</td>
<td>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).</td>
</tr>
<tr>
<td>YDF [JT]</td>
<td></td>
</tr>
</tbody>
</table>

\textsuperscript{a} See chapter "RUNNING THE EDIT MODULE" starting on page 8-1 for further discussion.  
\textsuperscript{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.  
\textsuperscript{c} You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.  
\textsuperscript{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:

\[ \text{XDF}(i) \times \text{YDF}(j) \]
Edit Cross-Section Types by Position and Name

<table>
<thead>
<tr>
<th>CROSS-SECTION INPUT VIA ISOTXS or GRUPXS</th>
<th>CROSS-SECTION INPUT VIA ASCII TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>EDIT Position</strong></td>
</tr>
<tr>
<td>chi</td>
<td>1</td>
</tr>
<tr>
<td>nu-fission</td>
<td>2</td>
</tr>
<tr>
<td>total</td>
<td>3</td>
</tr>
<tr>
<td>absorption</td>
<td>4</td>
</tr>
<tr>
<td>(n,p)</td>
<td>5</td>
</tr>
<tr>
<td>(n,d)</td>
<td>6</td>
</tr>
<tr>
<td>(n,t)</td>
<td>7</td>
</tr>
<tr>
<td>(n,alpha)</td>
<td>8</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>9</td>
</tr>
<tr>
<td>(n,gamma)</td>
<td>10</td>
</tr>
<tr>
<td>fission</td>
<td>11</td>
</tr>
<tr>
<td>transport</td>
<td>12</td>
</tr>
</tbody>
</table>

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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSFE [NGROUP;M]</td>
<td>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.</td>
</tr>
<tr>
<td>RSFX [IT;M]^b</td>
<td>Response function x (or r) distribution for M functions.</td>
</tr>
<tr>
<td>RSFY [JT;M]</td>
<td>Response function y (or z) (or theta) distribution for M functions.</td>
</tr>
</tbody>
</table>

{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:

\[
RSFX(i,m) \times RSFY(j,m) \times RSFE(g,m)
\]
### Energy Group Collapse Specifications

(Optional)

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOLL [NBG]</td>
<td>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).</td>
</tr>
<tr>
<td>IGRPED</td>
<td>Print option on energy groups. Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>
Reaction Rate Summing
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICSUM [&lt;500 sums]</td>
<td>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).</td>
</tr>
<tr>
<td>IRSUMS [&lt;500 sums]</td>
<td>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.</td>
</tr>
</tbody>
</table>

Mass Inventories
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSED</td>
<td>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.</td>
</tr>
</tbody>
</table>
Power Normalization
{Optional}

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

\(^a\) Note that this normalization is meaningless if you are using the results of an adjoint run.
Miscellaneous Edit Items
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RZFLUX</td>
<td>Write the CCCC standard zone\textsuperscript{a} flux file RZFLUX or AZFLUX. 0/1 = no/yes.</td>
</tr>
<tr>
<td>RZMFLX</td>
<td>Write the code-dependent zone\textsuperscript{b} flux moments file RZMFLX or AZMFLUX. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EOUTF\textsuperscript{c}</td>
<td>ASCII output files control. Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-3</td>
<td>Write both EDTOGX (without scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>-2</td>
<td>Write EDTOGX file (without scalar fluxes).</td>
</tr>
<tr>
<td>0</td>
<td>Write neither file. (default)</td>
</tr>
<tr>
<td>1</td>
<td>Write EDTOUT file.</td>
</tr>
<tr>
<td>2</td>
<td>Write EDTOGX file (with scalar fluxes).</td>
</tr>
<tr>
<td>3</td>
<td>Write both EDTOGX (with scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>BYVOLP</td>
<td>Printed point reaction rates will have been multiplied by the mesh volume. 0/1 = no/yes.</td>
</tr>
<tr>
<td>AJED\textsuperscript{d}</td>
<td>Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. 0/1 = regular/adjoint.</td>
</tr>
<tr>
<td>FLUXONE</td>
<td>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.</td>
</tr>
</tbody>
</table>

\textsuperscript{a} RZFLUX and AXFLUX are organized by solver zones. \hfill
\textsuperscript{b} RZMFLX and AZMFLX are organized by solver zones. \hfill
\textsuperscript{c} See “ASCII File Output Capabilities (the EOUTF Parameter)” on page 8-15. \hfill
\textsuperscript{d} See “Adjoint Edits” on page 8-15.
Special Plot Linkage
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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**

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \chi )</td>
<td>CHI</td>
<td>fission spectrum</td>
</tr>
<tr>
<td>( v\sigma_f )</td>
<td>NUSIGF</td>
<td>effective nu-sigma-fission</td>
</tr>
<tr>
<td>( \sigma_t )</td>
<td>TOTAL</td>
<td>Total cross section</td>
</tr>
<tr>
<td>( \sigma_a )</td>
<td>ABS</td>
<td>absorption^a</td>
</tr>
<tr>
<td>(n,n)</td>
<td>MEND1</td>
<td>elastic scattering</td>
</tr>
<tr>
<td>(n,n')</td>
<td>MEND2</td>
<td>inelastic scattering</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>MEND3</td>
<td>n,2n scattering</td>
</tr>
<tr>
<td>(n,3n)</td>
<td>MEND4</td>
<td>n,3n scattering</td>
</tr>
<tr>
<td>(n,\gamma)</td>
<td>MEND5</td>
<td>gamma production</td>
</tr>
<tr>
<td>(n,\alpha)</td>
<td>MEND6</td>
<td>alpha production</td>
</tr>
<tr>
<td>(n,p)</td>
<td>MEND7</td>
<td>proton production</td>
</tr>
<tr>
<td>(n,f)</td>
<td>MEND8</td>
<td>direct fission</td>
</tr>
<tr>
<td>(n,n')f</td>
<td>MEND9</td>
<td>second-chance fission</td>
</tr>
<tr>
<td>(n,2n)f</td>
<td>MEND10</td>
<td>third-chance fission</td>
</tr>
<tr>
<td>(n,F)</td>
<td>N-FISS</td>
<td>([(n,F) = (n,f) + (n,n')f + (n,2n)f])</td>
</tr>
<tr>
<td>( \chi_p )</td>
<td>MEND12</td>
<td>prompt fission spectrum</td>
</tr>
<tr>
<td>( \chi_t )</td>
<td>MEND13</td>
<td>total fission spectrum</td>
</tr>
</tbody>
</table>

^a. \( \sigma_a \) 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


APPENDIX A: SAMPLE INPUT

Sample Problem 1: Standard $k_{\text{eff}}$ 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

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

1. 2 0 0
   sample problem for twodant user's guide
2. standard k calculation, all input by means of card-images
3. geometry F 2
4. cross sections (2) group, isotropic scatter
5. mixing isotopic data on cards, los alamos (dtf) format
6. fuel, and sodium
7. materials assigned to make zones named core
8. solver - card input supplied
9. edits - zone edits for resident materials
10. 
11. 
12. 
13. 
14. 
15. * * * block i * * *
   j=0 1=0 3=0 4=0 5=0 6=0 7=0 8=0 9=0 10=0
16. zm=na1 zm=na2 zm=na3 zm=na4 zm=na5 zm=na6 zm=na7 zm=na8 zm=na9 zm=na10
17. zm=na11 zm=na12 zm=na13 zm=na14 zm=na15 zm=na16 zm=na17 zm=na18 zm=na19 zm=na20
18. zm=na21 zm=na22 zm=na23 zm=na24 zm=na25 zm=na26 zm=na27 zm=na28 zm=na29 zm=na30
19. * * * block ii (geometry) * * *
20. zm=na1 zm=na2 zm=na3 zm=na4 zm=na5 zm=na6 zm=na7 zm=na8 zm=na9 zm=na10
21. zm=na11 zm=na12 zm=na13 zm=na14 zm=na15 zm=na16 zm=na17 zm=na18 zm=na19 zm=na20
22. zm=na21 zm=na22 zm=na23 zm=na24 zm=na25 zm=na26 zm=na27 zm=na28 zm=na29 zm=na30
23. 
24. * * * block iii (cross sections) * * *
25. lib=libdxp
26. lib=libdxp
27. sample 2 group libdxp cross sections
28. sample 2 group libdxp cross sections
29. sample 2 group libdxp cross sections
30. sample 2 group libdxp cross sections
31. sample 2 group libdxp cross sections
32. sample 2 group libdxp cross sections
33. sample 2 group libdxp cross sections
34. sample 2 group libdxp cross sections
35. sample 2 group libdxp cross sections
36. sample 2 group libdxp cross sections
37. sample 2 group libdxp cross sections
38. sample 2 group libdxp cross sections
39. sample 2 group libdxp cross sections
40. sample 2 group libdxp cross sections
41. sample 2 group libdxp cross sections
42. sample 2 group libdxp cross sections
43. sample 2 group libdxp cross sections
44. sample 2 group libdxp cross sections
45. sample 2 group libdxp cross sections
46. sample 2 group libdxp cross sections
47. sample 2 group libdxp cross sections
48. sample 2 group libdxp cross sections
49. sample 2 group libdxp cross sections
50. sample 2 group libdxp cross sections
51. sample 2 group libdxp cross sections
52. sample 2 group libdxp cross sections
53. sample 2 group libdxp cross sections
54. sample 2 group libdxp cross sections
55. sample 2 group libdxp cross sections
56. sample 2 group libdxp cross sections
57. 
58. **** end of cross section data ****
59. **** note that there is no terminal '*' since the cross sections are
60. in los alamos (dtf) format (ifEnd=0) ****
61. 
62. * * * block iv (mixing) * * *
63. material steel fe 0.01 cr 0.01 ni 0.01
64. fuel pu-239 0.013, pu-236 0.013, '16' 0.0142
65. sodium na-23 0.025
66. assign core fuel 0.35 sodium 0.4 steel 0.25
67. reflect sodium 0.7 steel 0.3
68. 
69. 
70. * * * block v (solver) * * *
71. lset 1 lset 2 lset 3
72. rcset 1 rcset 2 rcset 3
73. chi 0.6 0.4 0.7 0.3
74. 
75. 
76. * * * block vi (edits) * * *
77. med 1 residual t
78. 
79. TWODANT USER'S GUIDE Version 3.0 3-79
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Output Listing

*****************************************************************************
* case title  
*****************************************************************************
* key start case input  
*****************************************************************************
  2 rhad number of title cards to follow  
  0 notcy 0/1 no/yes suppress or-line terminal output  
  0 callst 0/1 no/yes suppress input listing  
*****************************************************************************
* sample problem for twodant user's guide  
* standard x calculation, all input by means of card images  
*****************************************************************************
* key end block i read*  
*****************************************************************************

*****************************************************************************
* ...block i - controls and dimensions...  
*****************************************************************************
* ...dimensions (array name = dimens)...  
7 igem 6/7/9/11 x/y/r-z/triangles/x-theta  
2 rgroup number of energy groups  
4 igr angular quadrature order  
7 ruse number of input listeges (from ionos, groups, or cards)  
3 tnp number of permanent materials  
2 vname number of zones  
2 zin number of coarse mesh z intervals  
2 jq number of fine mesh intervals  
3 3g number of coarse mesh y intervals  
30 jj number of fine mesh y intervals  
...storage...  
modcon 140000  
maxcon 40000  
*****************************************************************************
* key end block ii read-geoms*  
*****************************************************************************
7 igem 1/2/3/6/7/8/9/11/14

*****************************************************************************
**APPENDIX A: SAMPLE INPUT**

Sample Problem 1: Output Listing

---

**...block iii - cross section library...**

**Library source...**

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

<table>
<thead>
<tr>
<th>group</th>
<th>chi</th>
<th>vel</th>
<th>lower bound</th>
<th>upper bound</th>
<th>group</th>
<th>chi</th>
<th>vel</th>
<th>lower bound</th>
<th>upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>2</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
</tbody>
</table>

**...energy structure...**

**last neutron group (ing) is number 2**

**edit position names...**

<table>
<thead>
<tr>
<th>card</th>
<th>position name</th>
<th>position edrane position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>chi</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>def</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>total</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>abs</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>edit</td>
<td>1</td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th>isotopic isotope number name order</th>
<th>header card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. o-16 p0 - oxygen-16 (o-16) sample 2 group</td>
<td>ref cross sections</td>
</tr>
<tr>
<td>2. na-23 p0 - sodium (na-23) sample 2 group</td>
<td>ref cross sections</td>
</tr>
<tr>
<td>3. fe p0 - iron (fe) sample 2 group</td>
<td>ref cross sections</td>
</tr>
<tr>
<td>4. cr p0 - chromium (cr) sample 2 group</td>
<td>ref cross sections</td>
</tr>
<tr>
<td>5. ni p0 - nickel (ni) sample 2 group</td>
<td>ref cross sections</td>
</tr>
<tr>
<td>6. pu-239 p0 - plutonium (pu-239) sample 2 group</td>
<td>ref cross sections</td>
</tr>
<tr>
<td>7. u-238 p0 - uranium (u-238) sample 2 group</td>
<td>ref cross sections</td>
</tr>
</tbody>
</table>
**APPENDIX A: SAMPLE INPUT**

**Sample Problem 1: Output Listing**

```plaintext

<table>
<thead>
<tr>
<th>mix</th>
<th>comp</th>
<th>density</th>
<th>comp</th>
<th>density etc.</th>
</tr>
</thead>
</table>

**metals**

1. steel  fe  5.00000E-02, cr  1.60000E-02, ni  1.00000E-02,
2. fuel  pu-239  1.20000E-02, u-238  1.03000E-02, o-16  4.14000E-02,
3. sodium  na-23  2.50000E-02,
4. sodium  na-23

**design**

1. comp  fuel  3.50000E-01, sodium  4.00000E-01, steel  2.50000E-01,
2. comp  fuel  3.00000E-01, steel  1.00000E-01,
3. comp  fuel  3.00000E-01, steel  1.00000E-01,
4. comp  fuel  3.00000E-01, steel  1.00000E-01,
```

---

*Key start mix card as*

*Key end mix card as*

*Key start block v read-solver*

*Key end block v read-solver*

*Key end block v read-edit*

*Key end input module*
Sample Problem 1: Output Listing

```
...block v -- solver input...

أعضاء c++ input (continued)

...miscellaneous parameters (array name = misc)

0.000E+00 0.000E+00 bhint building height
1.000E+00 0.000E+00 nmax normalization factor
0 0 influex 0/1 no/yes - read input flux form file rflux (eflux for adjoint)
0 0 inems 0/1 no/yes - read input source from file finsrc
0 00000 source number of saves/traill for flux or option
0 25 fout number of trials for flux or option
0 1 lquad -3/2/1/3/5 - source of quadrature constants (default=1)
  -3 source file
  -2 hybrid product set (triangular arrangement)
  1 old hybrid built-in set
  2 product set (rectangular arrangement)
  3 card input

...output controls (array name = solout)

1 fluxp 0/1/2 none/irregular/all moments - flux print
2 geselp 0/1/2 none/principal/all - macroscopic cross section print
0 inexc 0/1 no/yes - print final fission source rate
0 srcp 0/1/2/3 no/as read/normailized/both - print inhomogeneous source
0 copy 0/1 no/yes - print angular fluxes
0 rfluxp 0/1 no/yes - write angular fluxes to file rflux (eflux for adjoint)
0 mflux 0/1 no/yes - write flux moments to file mflux
0 halp 0/1 no/yes - print source mesh balances

...parameters inferred from input arrays...
0 incl 0/1/2 none/one chi/none chi
0 idensx 0/1 - nonp density vector/full matrix
0 idensy 0/1/2 no/yes - use y density vector
0 igt - source integration
0 incm number of source moments input
0 incm number of source moments input
```

APPENDIX A: SAMPLE INPUT

Sample Problem 1: Output Listing

---

```
...block v -- solver input...

...required input (array name = solin)...

1 1 lctx 0/1/2/3/4 - type of calculation
  0 homogeneous source
  1 k-effective
  2 alpha or time absorption search
  3 concentration search
  4 delta (i.e. dimension) search
0 0 lctx legendre order of scattering
0 0 lth 0/1 - direct/adjoint - mode of calculation (default-direct)
0 1 lth 0/1/2 - left boundary condition
0 0 lth 0/1/2 - right boundary condition
0 0 lth 0/1/2 - top boundary condition
0 0 lth 0/1/2 - bottom boundary condition
0 0 lth 0/1/2 - left boundary condition
vacuum/reflective/pervious/White
vacuum/reflective/pervious/White
vacuum/reflective/pervious/White
vacuum/reflective/pervious/White
vacuum/reflective/pervious/White

...convergence controls (array name = iter)...
0.000E+00 0.000E+00 spcl inner iteration convergence criterion (default=0.0001)
0 1 lcti maximum number of inner iterations per group until fission source is near
  convergence, i.e. lcti is near convergence (default=1)
0 30 lcti maximum number of inner iterations per group when close to fission source
  convergence (default=30)
0 20 lcte maximum number of outer iterations (default=20)
0 0 lcte iteration time limit (seconds)

...miscellaneous parameters...
0 0 recips set, fissions zero when source calculation (0/1 no/yes)
0 0 setf write in-going angular boundary flux to file abflux (0/1 no/yes)
0 0 setr write out-going angular boundary flux to file abflux (0/1 no/yes)
0 0 setu write out-going angular boundary flux to file abflux (0/1 no/yes)
0 0 seto write net-going angular boundary flux to file abflux (0/1 no/yes)
0 0 setf flux face option (0/1/2 no/analytic/try tracing)
```

---

TWODANT USER'S GUIDE  Version 3.0 3-83
Sample Problem 1: Output Listing

| 0 | source number of source moments input |
| 0 | latent number of source moments input |
| 0 | qgl -1/0/1/2 isotropic/mixed/all angles/velocity -left boundary source |
| 0 | qgr -1/0/1/2 isotropic/mixed/all angles/velocity -right boundary source |
| 0 | qgb -1/0/1/2 isotropic/mixed/all angles/velocity -bottom boundary source |

...parameters from block i...

| 7 | igem 677/11 x-y-z=r-theta |
| 2 | ngroup number of energy groups |
| 4 | lem angular quadrature order |
| 3 | st number of permanent materials |
| 2 | nume number of zones |
| 2 | qm number of coarse mesh x intervals |
| 2 | qy number of coarse mesh y intervals |
| 2 | 80 | number of fine mesh x intervals |
| 2 | 81 | number of fine mesh y intervals |
### APPENDIX A: SAMPLE INPUT

#### Sample Problem 1: Output Listing

```plaintext
/* key start zones map */

ZONE CROSS SECTION = SUM OVER NELS IN THE ZONE OF (REAL CROSS SECTION)*C0 = C1*CROSS 

with:

CROSS = 0.000000E+00

<table>
<thead>
<tr>
<th>entry</th>
<th>zone</th>
<th>material</th>
<th>c0</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>core</td>
<td>3.500000E-01</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>fuel</td>
<td>4.000000E-01</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>sodium</td>
<td>2.500000E-01</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>sodium</td>
<td>7.000000E-01</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>steel</td>
<td>3.000000E-01</td>
<td>0.000000E+00</td>
</tr>
</tbody>
</table>

/* key start storage map */

TOTAL SCM REQUIRED FOR THIS PROBLEM IS 10979
MAXIMUM SCM AVAILABLE (MACH) IS 40000

SCM REQUIRED FOR DIFFUSION IS 10978

SCM storage summary...

| SCM selected for this problem | 12689 |
| SCM required for all quantities in core is | 12689 |

SCM storage required for quantities in core is 12689
SCM storage required for all quantities on disk is 8288

12689 words SCM required 12689

/* zero flux */

** key start on constants **

<table>
<thead>
<tr>
<th>constant for groups 1 to 2</th>
<th>nu</th>
<th>eta</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3015878E+00</td>
<td>0.3015878E+00</td>
<td>0.3015878E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.3015878E+00</td>
<td>0.3015878E+00</td>
<td>0.3015878E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.3015878E+00</td>
<td>0.3015878E+00</td>
<td>0.3015878E+00</td>
</tr>
</tbody>
</table>

/* key start zones map */

ZONE BY COARSE MESH, ORIGIN AT LOWER LEFT.
row = number of fine intervals per coarse mesh.

<table>
<thead>
<tr>
<th>row</th>
<th>m</th>
<th>r</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>75.000</td>
<td>50</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3.5</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>15.000</td>
<td>0.000000</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>column</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
```

---

**TWODANT USER'S GUIDE Version 3.0**

**3-85**
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Output Listing

...cross section related data from file macros 11/28/84 version 1...

1 steel 2 fuel 3 sodium

...cross sections for legendre orders up to p7...

******************************************************
group 1 ******************************************************

...principal cross sections...

<table>
<thead>
<tr>
<th>zone</th>
<th>chi</th>
<th>nu_fission</th>
<th>total</th>
<th>absorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>core</td>
<td>6.00000E-01</td>
<td>2.5848E-02</td>
<td>1.2494E-01</td>
</tr>
<tr>
<td>2</td>
<td>refl</td>
<td>7.00000E-01</td>
<td>0.0000E+00</td>
<td>8.371E-02</td>
</tr>
</tbody>
</table>

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

zone order first grup cross sections
1 0 1 9.0947E-02
2 0 1 6.9979E-02

******************************************************
group 2 ******************************************************

...principal cross sections...

<table>
<thead>
<tr>
<th>zone</th>
<th>chi</th>
<th>nu_fission</th>
<th>total</th>
<th>absorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>core</td>
<td>4.00000E-01</td>
<td>1.7304E-02</td>
<td>2.7628E-01</td>
</tr>
<tr>
<td>2</td>
<td>refl</td>
<td>3.00000E-01</td>
<td>0.0000E+00</td>
<td>1.8568E-01</td>
</tr>
</tbody>
</table>

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

zone order first grup cross sections
1 0 2 2.9714E-01 2.4203E-02
2 0 2 1.8568E-01 1.5103E-02

******************************************************
**APPENDIX A: SAMPLE INPUT**

Sample Problem 1: Output Listing

---

### **Iteration Controls and Criteria**

<table>
<thead>
<tr>
<th><strong>Criterion</strong></th>
<th><strong>Test Quantity</strong></th>
<th><strong>Value</strong></th>
<th><strong>Action if Value Exceeded</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>inner</code></td>
<td>inner iteration count until near lambda</td>
<td>1</td>
<td>terminates inner</td>
</tr>
<tr>
<td></td>
<td>inner iteration count when near lambda</td>
<td>30</td>
<td>terminates inner</td>
</tr>
<tr>
<td><code>epsi</code></td>
<td>fractional power flux change per inner</td>
<td>1.000E-04</td>
<td>does another inner</td>
</tr>
</tbody>
</table>

### **Transport Inners**

<table>
<thead>
<tr>
<th><strong>Criterion</strong></th>
<th><strong>Test Quantity</strong></th>
<th><strong>Value</strong></th>
<th><strong>Action if Value Exceeded</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cmlzd</code></td>
<td>sub-inner iteration count</td>
<td>40</td>
<td>terminates sub-inners</td>
</tr>
<tr>
<td><code>eps</code></td>
<td>diffusion lambda-1.0 (see note below)</td>
<td>1.000E-04</td>
<td>does another sub-inner</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th><strong>Criterion</strong></th>
<th><strong>Test Quantity</strong></th>
<th><strong>Value</strong></th>
<th><strong>Action if Value Exceeded</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aim</code></td>
<td>outer iteration count</td>
<td>20</td>
<td>quits with error message</td>
</tr>
<tr>
<td><code>epsi</code></td>
<td>transport lambda-1.0</td>
<td>1.000E-04</td>
<td>does another outer</td>
</tr>
</tbody>
</table>
### APPENDIX A: SAMPLE INPUT

Sample Problem 1: Output Listing

---

...flux and eigenvalue convergence as monitored by twodant...

---

**Key start iteration monitor**

---

<table>
<thead>
<tr>
<th>cpu time</th>
<th>diffusion</th>
<th>k-eff</th>
<th>(sec)</th>
<th>no.</th>
<th>inner sub-outer</th>
<th>eigenvalue</th>
<th>lambda-1</th>
<th>max ptdise</th>
<th>max ptdise</th>
<th>inners</th>
<th>flux change</th>
<th>flux change converged</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.81</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.38088580</td>
<td>5.79316765</td>
<td>0.00000020</td>
<td>1.48189520</td>
<td>0.938139520</td>
<td><strong>nom</strong></td>
<td></td>
</tr>
<tr>
<td>3.88</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0.38088580</td>
<td>5.79316765</td>
<td>0.00000020</td>
<td>1.48189520</td>
<td>0.938139520</td>
<td><strong>nom</strong></td>
<td></td>
</tr>
<tr>
<td>3.97</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0.38088580</td>
<td>5.79316765</td>
<td>0.00000020</td>
<td>1.48189520</td>
<td>0.938139520</td>
<td><strong>nom</strong></td>
<td></td>
</tr>
</tbody>
</table>

---

inner iteration summary for outer iteration no. 4

---

<table>
<thead>
<tr>
<th>cpu time</th>
<th>diffusion</th>
<th>k-eff</th>
<th>(sec)</th>
<th>no.</th>
<th>inner sub-outer</th>
<th>eigenvalue</th>
<th>lambda-1</th>
<th>max ptdise</th>
<th>max ptdise</th>
<th>inners</th>
<th>flux change</th>
<th>flux change converged</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.07</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0.38063020</td>
<td>5.95137950</td>
<td>0.99426505</td>
<td>0.89276050</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

SSSSSS all convergence criteria satisfied $SSSSS$

---

particle balance = -6.397378E-10 total inners all outers = 10

---

---

**Key start balance table**

---

<table>
<thead>
<tr>
<th>gp</th>
<th>source</th>
<th>fission source</th>
<th>in scatter</th>
<th>self scatter</th>
<th>cut scatter</th>
<th>not leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>6.00000000E+00</td>
<td>-1.0659481E-14</td>
<td>1.4713377E+00</td>
<td>3.7518525E-01</td>
<td>1.4796478E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>4.00000000E+00</td>
<td>3.7518525E-01</td>
<td>1.4796478E-01</td>
<td>3.7518525E-01</td>
<td>3.7518525E-01</td>
</tr>
<tr>
<td>tot</td>
<td>0.00000000E+00</td>
<td>1.00000000E+00</td>
<td>3.7518525E-01</td>
<td>1.4796478E-01</td>
<td>3.7518525E-01</td>
<td>3.7518525E-01</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>gp</th>
<th>absorption</th>
<th>particle balance</th>
<th>right leakage</th>
<th>horizontal leakage</th>
<th>top leakage</th>
<th>vertical leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.199541E+01</td>
<td>7.099176E-08</td>
<td>7.165579E-02</td>
<td>7.165579E-02</td>
<td>2.554098E-02</td>
<td>4.109798E-02</td>
</tr>
<tr>
<td>2</td>
<td>4.423574E-01</td>
<td>-6.339102E-08</td>
<td>2.061316E-01</td>
<td>2.061316E-01</td>
<td>6.340426E-02</td>
<td>1.268098E-01</td>
</tr>
<tr>
<td>tot</td>
<td>5.573898E-01</td>
<td>-6.397378E-10</td>
<td>2.770833E-01</td>
<td>2.770833E-01</td>
<td>8.453923E-02</td>
<td>1.690782E-01</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>gp</th>
<th>left leakage</th>
<th>bottom leakage</th>
<th>reprod spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000000E+00</td>
<td>2.154889E-02</td>
<td>3.189515E-02</td>
</tr>
<tr>
<td>2</td>
<td>0.00000000E+00</td>
<td>6.340426E-02</td>
<td>6.340426E-02</td>
</tr>
<tr>
<td>tot</td>
<td>0.00000000E+00</td>
<td>8.453923E-02</td>
<td>1.690782E-01</td>
</tr>
</tbody>
</table>

Timing for sweep and DMA: tastics: 0.105 and time: 0.139 and tcache: 0.000 seconds.

---

Integral summary information

---

<table>
<thead>
<tr>
<th>integral</th>
<th>k-eff</th>
<th>summary (integral-k-eff)</th>
<th>9.89630325E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>integral-source-1</td>
<td>neutron</td>
<td>0.00000000E+00</td>
<td></td>
</tr>
<tr>
<td>integral-fission-1</td>
<td>neutron</td>
<td>1.00000000E+00</td>
<td></td>
</tr>
<tr>
<td>integral-in-scatt-1</td>
<td>neutron</td>
<td>3.7518525E-01</td>
<td></td>
</tr>
<tr>
<td>integral-self-scatt-1</td>
<td>neutron</td>
<td>1.6497945E-01</td>
<td></td>
</tr>
<tr>
<td>integral-cut-scatt-1</td>
<td>neutron</td>
<td>3.7518525E-01</td>
<td></td>
</tr>
<tr>
<td>integral-reset-scatt-1</td>
<td>neutron</td>
<td>4.472131E-01</td>
<td></td>
</tr>
<tr>
<td>integral-absorption-1</td>
<td>neutron</td>
<td>5.827666E-01</td>
<td></td>
</tr>
<tr>
<td>integral-right-scatt-1</td>
<td>neutron</td>
<td>2.773065E-01</td>
<td></td>
</tr>
<tr>
<td>integral-horizontal-scatt-1</td>
<td>neutron</td>
<td>2.773065E-01</td>
<td></td>
</tr>
<tr>
<td>integral-top-scatt-1</td>
<td>neutron</td>
<td>8.453923E-02</td>
<td></td>
</tr>
<tr>
<td>integral-vertical-scatt-1</td>
<td>neutron</td>
<td>1.690782E-01</td>
<td></td>
</tr>
</tbody>
</table>
## APPENDIX A: SAMPLE INPUT

### Sample Problem 1: Output Listing

<table>
<thead>
<tr>
<th>Component number</th>
<th>Output Listing</th>
</tr>
</thead>
<tbody>
<tr>
<td>mesh 1</td>
<td>mesh 2</td>
</tr>
<tr>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>30</td>
<td>1.590295x-05</td>
</tr>
<tr>
<td>31</td>
<td>2.542364x-05</td>
</tr>
<tr>
<td>32</td>
<td>3.186192x-05</td>
</tr>
<tr>
<td>33</td>
<td>4.041169x-05</td>
</tr>
<tr>
<td>34</td>
<td>4.814608x-05</td>
</tr>
<tr>
<td>35</td>
<td>5.990590x-05</td>
</tr>
<tr>
<td>36</td>
<td>4.041169x-05</td>
</tr>
<tr>
<td>37</td>
<td>4.814608x-05</td>
</tr>
<tr>
<td>38</td>
<td>5.990590x-05</td>
</tr>
</tbody>
</table>

...
**APPENDIX A: SAMPLE INPUT**

**Sample Problem 1: Output Listing**

<table>
<thead>
<tr>
<th>Component Number</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>6.2600150-05</td>
<td>6.199768-05</td>
<td>6.104852-05</td>
<td>5.976420-05</td>
<td>5.813139-05</td>
<td>5.639780-05</td>
<td>5.389000-05</td>
<td>5.122292-05</td>
</tr>
<tr>
<td></td>
<td>1.085313-04</td>
<td>1.068526-04</td>
<td>1.042540-04</td>
<td>1.020430-04</td>
<td>0.952600-04</td>
<td>0.888960-04</td>
<td>0.826840-04</td>
<td>0.765570-04</td>
</tr>
<tr>
<td></td>
<td>1.272799-05</td>
<td>1.290351-05</td>
<td>1.305351-05</td>
<td>1.323451-05</td>
<td>1.373551-05</td>
<td>1.423651-05</td>
<td>1.473751-05</td>
<td>1.523851-05</td>
</tr>
<tr>
<td></td>
<td>2.925165-06</td>
<td>3.029119-06</td>
<td>3.134069-06</td>
<td>3.239019-06</td>
<td>3.439069-06</td>
<td>3.639119-06</td>
<td>3.839169-06</td>
<td>4.039219-06</td>
</tr>
<tr>
<td></td>
<td>2.403447-06</td>
<td>2.385947-06</td>
<td>2.368447-06</td>
<td>2.350947-06</td>
<td>2.333447-06</td>
<td>2.315947-06</td>
<td>2.308447-06</td>
<td>2.300947-06</td>
</tr>
<tr>
<td></td>
<td>4.619618-05</td>
<td>4.731781-05</td>
<td>4.843954-05</td>
<td>4.956117-05</td>
<td>5.068280-05</td>
<td>5.180443-05</td>
<td>5.292606-05</td>
<td>5.404769-05</td>
</tr>
<tr>
<td></td>
<td>5.149772-05</td>
<td>5.103169-05</td>
<td>5.022860-05</td>
<td>4.908360-05</td>
<td>4.795860-05</td>
<td>4.683360-05</td>
<td>4.570860-05</td>
<td>4.458360-05</td>
</tr>
<tr>
<td></td>
<td>5.331167-05</td>
<td>5.373554-05</td>
<td>5.416041-05</td>
<td>5.458534-05</td>
<td>5.501031-05</td>
<td>5.543528-05</td>
<td>5.586025-05</td>
<td>5.628522-05</td>
</tr>
<tr>
<td></td>
<td>5.448443-05</td>
<td>5.454443-05</td>
<td>5.459443-05</td>
<td>5.464443-05</td>
<td>5.469443-05</td>
<td>5.474443-05</td>
<td>5.479443-05</td>
<td>5.484443-05</td>
</tr>
<tr>
<td></td>
<td>5.696562-05</td>
<td>5.724468-05</td>
<td>5.752374-05</td>
<td>5.779180-05</td>
<td>5.806086-05</td>
<td>5.833992-05</td>
<td>5.861898-05</td>
<td>5.889804-05</td>
</tr>
<tr>
<td></td>
<td>5.915772-05</td>
<td>5.903169-05</td>
<td>5.890565-05</td>
<td>5.878062-05</td>
<td>5.865559-05</td>
<td>5.853056-05</td>
<td>5.840553-05</td>
<td>5.828050-05</td>
</tr>
<tr>
<td></td>
<td>6.2000150-05</td>
<td>6.199768-05</td>
<td>6.104852-05</td>
<td>5.976420-05</td>
<td>5.813139-05</td>
<td>5.639780-05</td>
<td>5.389000-05</td>
<td>5.122292-05</td>
</tr>
</tbody>
</table>

**Flux Components for Group 2**

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.022445-05</td>
<td>4.022445-05</td>
<td>4.022445-05</td>
<td>4.022445-05</td>
<td>4.022445-05</td>
<td>4.022445-05</td>
<td>4.022445-05</td>
<td>4.022445-05</td>
</tr>
</tbody>
</table>

---

**TWODANT USER'S GUIDE**

Version 3.0
**APPENDIX A: SAMPLE INPUT**

**Sample Problem 1: Output Listing**

<table>
<thead>
<tr>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
<th>r mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>2.34445E-05</td>
<td>4.89988E-05</td>
<td>4.35259E-05</td>
<td>2.77378E-05</td>
<td>2.40342E-05</td>
<td>1.27555E-05</td>
<td>1.10992E-05</td>
<td>1.37978E-05</td>
<td>1.49117E-05</td>
<td>1.58950E-05</td>
<td>1.69965E-05</td>
<td>1.82987E-05</td>
</tr>
</tbody>
</table>

| interface file filler written with the fission source. |
| fission neutron distribution |

<table>
<thead>
<tr>
<th>*Way start fission source</th>
<th>component number 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>r mesh</td>
<td>r mesh</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>2.34445E-05</td>
<td>4.89988E-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TWODANT USER'S GUIDE</th>
<th>Version 3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-91</td>
<td></td>
</tr>
</tbody>
</table>
## APPENDIX A: SAMPLE INPUT

### Sample Problem 1: Output Listing

<table>
<thead>
<tr>
<th>Mesh</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>Ux</th>
<th>Uy</th>
<th>Uz</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...interface file reflux written.

---

### Parameters:
- 3-92 TWODANT USERS GUIDE Version 3.0
Sample Problem 1: Output Listing

...interface file name written...

twodant iteration time, mins 6.96265-02
Sample Problem 1: Output Listing

******************************************************************************
** 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 (balbal=0) **
* or *
** transport correction (wire, ctnel, cassce, or bhe) **
** will NOT be reflected in edits **
******************************************************************************

** input control integers... **
*
0 pted 0/1  no/yes - point edits desired
1 smed 0/1  no/yes - zone edits desired
0 ajed 0/1  direct/adjoint edit (use refill/reflux file)
0 igsped 0/1/2/3  print totals only/print broad groups only/some as 1/print all groups and totals
0 byg0ps 0/1  no/yes - multiply point reaction rates by mesh volumes
0 refill 0/1  no/yes - write the refill file (zone average flux file)
0 refdx 0/1  no/yes - write the refdx file (zone average flux moments file)

** floating parameters... **
*
0.0000000E+00  power  0/p  no/normalize all results, including flux files, to p megawatts
2.1000000E+02  neper  n ev per fission (default: 210 ev)

** energy related edit information... **
*
2 number of fine neutron 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)
******************************************************************************
APPENDIX A: SAMPLE INPUT

Sample Problem 1: Output Listing

******************************************************************************
...edit output...
******************************************************************************

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

******************************************************************************
...description of cross section edits...
******************************************************************************

isotope no. material no. constituent no.
--- ------ ------ 
-none- readit -none-

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

type position
--- ----
chi     1
rusigf  2
total   3
abs     4
edit   5

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

******************************************************************************
 reaction rates 
******************************************************************************

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

* resident macroscopic...

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>chi</th>
<th>rusigf</th>
<th>total</th>
<th>abs</th>
<th>edit</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2432E+04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>8.7785E-01</td>
<td>2.2068E-03</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5.3914E+04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>3.5978E-01</td>
<td>9.1957E-04</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.7218E+05</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.2164E-01</td>
<td>3.3953E-01</td>
<td>3.1164E-01</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.5904E+05</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>2.6966E-01</td>
<td>6.8954E-03</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4.2412E+04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>8.7785E-01</td>
<td>2.2068E-03</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5.3914E+04</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>3.5978E-01</td>
<td>9.1957E-04</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>sum</td>
<td>4.7713E+05</td>
<td>0.00000E+00</td>
<td>9.8067E-02</td>
<td>1.7422E-01</td>
<td>5.5279E-01</td>
<td>3.1164E-01</td>
<td></td>
</tr>
</tbody>
</table>

******************************************************************************

TWODANT USER'S GUIDE  Version 3.0  3-95
run highlights

---

all modules are tentatively go:
interface file gender written.
cross sections from cards.
interface mg ty file written.
interface file aged written.
x files moved, read written.
left boundary condition overridden.
interface file scaling written.
interface file edit written.
start solver execution.
all convergence criteria met.
fixes written with fixmixes.
interface file rflux written.
interface file encore written.
start edit execution.
edit completed.
---

storage and timing history

---

<table>
<thead>
<tr>
<th>module</th>
<th>lca</th>
<th>lca</th>
<th>lca</th>
<th>cpu</th>
<th>lcpu</th>
<th>smap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.4</td>
<td>0.4</td>
</tr>
<tr>
<td>101</td>
<td>8539</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>102</td>
<td>1043</td>
<td>40000</td>
<td>10</td>
<td>140000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>103</td>
<td>50</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>2065</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>181</td>
<td>40000</td>
<td>80</td>
<td>140000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>108</td>
<td>592</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>4629</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>112</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>10997</td>
<td>40000</td>
<td>12689</td>
<td>140000</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>201</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>203</td>
<td>927</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>204</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>205</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>206</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>207</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>208</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>210</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>211</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>301</td>
<td>7481</td>
<td>40000</td>
<td>61</td>
<td>140000</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>302</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

... 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.
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/Sn 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/Sn 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

...Listing of cards in the input stream...

1. 1 0
2. sample envrproblm for package user's guide
3. *********************** BLOCK I ***********************
4. iipem= 2 nump= 2 ian=4
5. limit 1.25 3 pos=40
6. name=x mt=2 nts=2
7. t
8. *********************** BLOCK II ***********************
9. xwin=5 2 18 ywin=40
10. xwin=5 5 7 25
11. yen=50. 50.
12. xcen=0 1.2
13. *********************** BLOCK III ***********************
14. lib=cdulp
15. msw=1 thr=3 lhi=4 tinit=1
16. name=wall shield
17. t
18. wall, p0
19. 0.0 0.0 0.0 0.75 0.75 0.0 0.0
20. 0.22 0.0 0.0 1.14 0.92 0.0 0.0
21. wall, pl
22. 0.0 0.0 0.0 0.03 0.0 0.0
23. 0.0 0.0 0.0 0.01 0.0 0.0
24. shield, p0
25. 0.0 0.0 0.0 0.04 0.81 0.0 0.0
26. 0.0 0.0 0.0 1.32 1.14 0.02 0.0
27. 0.18 0.0 0.0 0.0
28. shield, pl
29. 0.0 0.0 0.0 0.0 0.0 0.0
30. 0.0 0.0 0.0 0.0 0.0 0.0
31. *********************** BLOCK IV ***********************
32. *********************** BLOCK V ***********************
33. *********************** BLOCK VI ***********************
34. t
35. *********************** BLOCK VII ***********************
36. *********************** BLOCK VIII ***********************
37. *********************** BLOCK IX ***********************
38. *********************** BLOCK X ***********************
39. *********************** BLOCK XI ***********************
40. *********************** BLOCK XII ***********************
Sample Problem 2: Output Listing

------------------------------------------------------------------------

    case title
------------------------------------------------------------------------

*  1 rhed  number of title cards to follow
  0 norty  0/1 no/yes suppress on-line terminal output
  0 noilst  0/1 no/yes suppress input listing

------------------------------------------------------------------------

*  sample en/mc problem for twodant user's guide

------------------------------------------------------------------------

*key end block i read*

------------------------------------------------------------------------

  ...block i - controls and dimensions...

------------------------------------------------------------------------

  ...dimensions (array name = dimen)...  
  7 igecm  6/7/9/11 x-y-z/xy-z/triangles/r-theta
  2 ngroup  number of energy groups
  2 ilen  angular quadrature order
  2 nuse  number of input isotopes (from isotos, groups, or cards)
  2 st  number of permanent materials
  2 name  number of zones
  2 im  number of coarse mesh x intervals
  2 it  number of fine mesh intervals
  2 gm  number of coarse mesh y intervals
  2 jf  number of fine mesh y intervals

  ...storage...
  1  read  140000
  2  read  40000

------------------------------------------------------------------------

*key end block ii read-gecm*

------------------------------------------------------------------------

  7 igecm  1/2/3/6/7/8/9/11/14

------------------------------------------------------------------------
**APPENDIX A: SAMPLE INPUT**

**Sample Problem 2: Output Listing**

```
*...block iii - cross section library...
*...
*library source...
libcheck
*...card library parameters (array name = cards)...
  1: recard maximum legendre order to be found in input cross sections
  2: lhm last position in cross section table
  3: lth position of total cross section
  4: ltn position of self scatter cross section
  0: ifido -1/0/1/1 - df(delf.0)/df/ixed fido/free fido library
  1: ifitcl 0/1 - ignores there is a title card before each table
  0: 2pl 0/1 - ignores library higher order scattering contains Z=1 factor
  0: sendas 0/1 - ignores save binary with file (filter=bellib)
  1: isekod 0/1 - full fido read/quick fido read (default=quick)
*...
*energy structure...

<table>
<thead>
<tr>
<th>group</th>
<th>chi</th>
<th>vel</th>
<th>lower bound</th>
<th>upper bound</th>
<th>group</th>
<th>chi</th>
<th>vel</th>
<th>lower bound</th>
<th>upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00008&gt;00</td>
<td>0.00008&gt;00</td>
<td>0.00008&gt;00</td>
<td>0.00008&gt;00</td>
<td>2</td>
<td>0.00008&gt;00</td>
<td>0.00008&gt;00</td>
<td>0.00008&gt;00</td>
<td>0.00008&gt;00</td>
</tr>
</tbody>
</table>

last neutron group(ing) is number 2

0: bales -1/0/1 - adjust absorption/no adjust self scatter to force xs balance
*...
*edit position names...
*card position edname position
  1: chi
  2: rustif
  3: total
  4: abs

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

<table>
<thead>
<tr>
<th>isotope</th>
<th>isotope</th>
<th>number</th>
<th>name</th>
<th>order</th>
<th>header card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. wall</td>
<td>p0</td>
<td>-</td>
<td>wall, p0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>p1</td>
<td>-</td>
<td>wall, pl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. shield</td>
<td>p0</td>
<td>-</td>
<td>shield, p0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>p1</td>
<td>-</td>
<td>shield, pl</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Sample Problem 2: Output Listing

<table>
<thead>
<tr>
<th>mix</th>
<th>comp</th>
<th>density</th>
<th>comp</th>
<th>density etc.</th>
</tr>
</thead>
</table>

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 solver*

*key end input module*
**APPENDIX A: SAMPLE INPUT**

Sample Problem 2: Output Listing

---

```plaintext
...block v -- solver input...

<table>
<thead>
<tr>
<th>raw input</th>
<th>as defaulted</th>
</tr>
</thead>
<tbody>
<tr>
<td>...required input (array name = solin)...</td>
<td></td>
</tr>
<tr>
<td>0 0 iact 0/1/2/3/4 - type of calculation</td>
<td></td>
</tr>
<tr>
<td>0 0 ihf 0/1 - fluxes/adjoints - mode of calculation (default-direct)</td>
<td></td>
</tr>
<tr>
<td>1 1 ihl 0/1/2/3 - left boundary condition</td>
<td></td>
</tr>
<tr>
<td>0 0 lbr 0/1/2/3 - right boundary condition</td>
<td></td>
</tr>
<tr>
<td>0 0 lbt 0/1/2/3 - top boundary condition</td>
<td></td>
</tr>
<tr>
<td>0 0 lbb 0/1/2/3 - bottom boundary condition</td>
<td></td>
</tr>
<tr>
<td>...convergence controls (array name = iter)...</td>
<td></td>
</tr>
<tr>
<td>1.0000-03 1.0000-03 apei inner iteration convergence criterion (default=0.0001)</td>
<td></td>
</tr>
<tr>
<td>0 1 ilit maximum number of inner iterations per group until fission sources is near convergence, i.e. lambda is near convergence (default=30)</td>
<td></td>
</tr>
<tr>
<td>0 20 olit maximum number of outer iterations (default=20)</td>
<td></td>
</tr>
<tr>
<td>0 0 fiti iteration time limit (seconds)</td>
<td></td>
</tr>
</tbody>
</table>
| ...miscellaneous parameters...
| 0 0 lnspf set fissions zero when source calculation (0/1 no/yes) |
| 0 0 renoc write in-going angular boundary flux to file afux (0/1 no/yes) |
| 0 0 renac write out-going angular boundary flux to file afux (0/1 no/yes) |
| 0 0 rennc write up-going angular boundary flux to file afux (0/1 no/yes) |
| 0 0 renbd write down-going angular boundary flux to file afux (0/1 no/yes) |
| 0 0 lufoe force option (0/1/2 no/analytic/ray tracing) |
```

---

3-106 TWODANT USER'S GUIDE Version 3.0
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

Definitions:
- **block v**: solver input (continued)
- **miscellaneous parameters**: (array name = misc)
- **output controls**: (array name = solout)
- **parameters inferred from input arrays**: 
- **parameters from block i**: 

TWODANT USER'S GUIDE  Version 3.0 3-107
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

-----------------------------------------------------------------------------------

...material assignments to zones...

*key start mats to zones *

-------------------------------------------------------------------------------

<table>
<thead>
<tr>
<th>zone</th>
<th>material</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>zone1</td>
</tr>
<tr>
<td>2</td>
<td>zone2</td>
</tr>
</tbody>
</table>

*key start storage map *

-------------------------------------------------------------------------------

sem storage summary...

- total sem required for this problem: 16000
- maximum sem available (if available): 40000
- sem required for transport: 15981
- sem required for diffusion: 15978

lcm storage summary...

- lcm selected for this problem: 30819
- maximum lcm specified (problem): 140000
- Storage required for all quantities in core is: 30819
  - source moments are in core, if O.K. to put in core, set MAXCUX -ge. 30819
  - diffusion phases are in core, if O.K. to put in core, set MAXCUX -ge. 30819
  - flux moments are in core, if O.K. to put in core, set MAXCUX -ge. 25494
  - similar source are in core, if O.K. to put in core, set MAXCUX -ge. 25494
  - diffusion fluxes are in core, if O.K. to put in core, set MAXCUX -ge. 22185
  - similar to fluxes are in core, if O.K. to put in core, set MAXCUX -ge. 22185
  - Storage required for all quantities on disk is: 19729

30819 words lcm required on disk

*key start flux *

**key eight constants**

- 4 constants for groups 1 to 2

<table>
<thead>
<tr>
<th>n</th>
<th>e</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.301639785600</td>
<td>0.301639785600</td>
</tr>
<tr>
<td>2</td>
<td>0.301639785600</td>
<td>0.301639785600</td>
</tr>
<tr>
<td>3</td>
<td>0.301639785600</td>
<td>0.301639785600</td>
</tr>
</tbody>
</table>

*key start zones map *

-------------------------------------------------------------------------------

zones by coarse mesh, origin at lower left.

z is number of fine intervals per coarse mesh.

-------------------------------------------------------------------------------

<table>
<thead>
<tr>
<th>row</th>
<th>m</th>
<th>50.0000</th>
<th>1 40</th>
<th>r</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>2 v</th>
<th>0.0000</th>
<th>1 2</th>
<th>5 7 18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-------------------------------------------------------------------------------

3-108  TWODANT USER'S GUIDE  Version 3.0
**APPENDIX A: SAMPLE INPUT**

**Sample Problem 2: Output Listing**

---

```plaintext
# Cross section related data from file npcones1k50139 version 1...

1 wall  2 shield

# Cross sections for legendre orders up to pl...

## Key start mac cross sections

### Group 1

**Principal cross sections**

<table>
<thead>
<tr>
<th>zone</th>
<th>chi</th>
<th>nu fission</th>
<th>total</th>
<th>absorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>7.000E-01</td>
<td>1.000E-02</td>
</tr>
<tr>
<td>2</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>8.000E-01</td>
<td>1.000E-02</td>
</tr>
</tbody>
</table>

**Scattering matrix**

(21+1 not included)

### Group 2

**Principal cross sections**

<table>
<thead>
<tr>
<th>zone</th>
<th>chi</th>
<th>nu fission</th>
<th>total</th>
<th>absorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>1.000E+00</td>
<td>2.000E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>1.000E+00</td>
<td>1.000E-01</td>
</tr>
</tbody>
</table>

**Scattering matrix**

(21+1 not included)
```

---

**TWODANT USER'S GUIDE**

Version 3.0

3-109
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

******************************************************************************
* sample input parameters...
******************************************************************************

1 accept 0/1 nolyes - mc option
8 mcpts max number of super-cuts
0 released initial random seed
10000 mcpts number of hist/trial
25 mcpts number of trials
1 left left mc boundary
5 right right mc boundary
0 top top mc boundary
0 bottom bottom mc boundary
1.0000e+00 subl boundary layer thickness
0.0000e+00 maxc weight cutoff 1
0.0000e+00 maxc weight cutoff 2
0.0000e+00 maxb disk source blurring

...parameters inferred from source arrays...

0 interc 0/1/2 dist/pt beam/pt source
0 intleft 0/1 nolyes - left beam source
0 intartic 0/1 nolyes - top beam source
0 intartic 0/1 nolyes - bottom beam source

******************************************************************************
* monte carlo setup information...
******************************************************************************

30959 words lcm required accept
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 

monte carlo fine mesh boundaries

<table>
<thead>
<tr>
<th>group</th>
<th>1bb</th>
<th>1bb</th>
<th>1bb</th>
<th>1bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the amount of the fixed source in the mc regplot is 1.0000e+00

******************************************************************************
* mc angular bin generation
******************************************************************************

nbin = 32
processing zone 1...
processing zone 2...
cpu time required = 6.6678-02 sec
**APPENDIX A: SAMPLE INPUT**

Sample Problem 2: Output Listing

---

**self scatter fem, source calculation**

<table>
<thead>
<tr>
<th>group 1</th>
<th>group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>nhie</td>
<td>mean</td>
</tr>
<tr>
<td>10020</td>
<td>4.02378E+00</td>
</tr>
<tr>
<td>20040</td>
<td>4.01296E+00</td>
</tr>
<tr>
<td>30060</td>
<td>4.01456E+00</td>
</tr>
<tr>
<td>40080</td>
<td>4.02096E+00</td>
</tr>
<tr>
<td>50100</td>
<td>4.02116E+00</td>
</tr>
<tr>
<td>60120</td>
<td>4.03096E+00</td>
</tr>
<tr>
<td>70140</td>
<td>4.03366E+00</td>
</tr>
<tr>
<td>80160</td>
<td>4.03196E+00</td>
</tr>
<tr>
<td>90180</td>
<td>4.03076E+00</td>
</tr>
<tr>
<td>100200</td>
<td>4.02296E+00</td>
</tr>
<tr>
<td>110220</td>
<td>4.02736E+00</td>
</tr>
<tr>
<td>120240</td>
<td>4.02696E+00</td>
</tr>
<tr>
<td>130260</td>
<td>4.01896E+00</td>
</tr>
<tr>
<td>140280</td>
<td>4.02016E+00</td>
</tr>
<tr>
<td>150300</td>
<td>4.02446E+00</td>
</tr>
<tr>
<td>160320</td>
<td>4.02366E+00</td>
</tr>
<tr>
<td>170340</td>
<td>4.02436E+00</td>
</tr>
<tr>
<td>180360</td>
<td>4.02236E+00</td>
</tr>
<tr>
<td>190380</td>
<td>4.01986E+00</td>
</tr>
<tr>
<td>200400</td>
<td>4.02036E+00</td>
</tr>
<tr>
<td>210420</td>
<td>4.02016E+00</td>
</tr>
<tr>
<td>220440</td>
<td>4.02336E+00</td>
</tr>
<tr>
<td>230460</td>
<td>4.02356E+00</td>
</tr>
<tr>
<td>240480</td>
<td>4.02366E+00</td>
</tr>
<tr>
<td>250500</td>
<td>4.02216E+00</td>
</tr>
</tbody>
</table>

Absolute error (1 std dev) in integral set is 7.279E-03

---

**mc coarse mesh tracklengths**

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Max = 0.0017 i = 1 j = 1

---

**mc coarse mesh leakages**

**top leakages**

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**bottom leakages**

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**right leakages**

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**left leakages**

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**tracks entering (coarse mesh)**

<table>
<thead>
<tr>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**TWODANT USER'S GUIDE Version 3.0 3-111**
## MC Source Calculation Particle Balance

<table>
<thead>
<tr>
<th>gp</th>
<th>source</th>
<th>files src(mc)</th>
<th>files src(sm)</th>
<th>in scatter(mc)</th>
<th>in scatter(sm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>2.7755E-17</td>
<td>1.0765E-01</td>
</tr>
<tr>
<td>tot</td>
<td>1.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>2.7755E-17</td>
<td>1.0765E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gp</th>
<th>self scatter</th>
<th>not leakage</th>
<th>absorption</th>
<th>cut scatter</th>
<th>particle balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.0223E+00</td>
<td>8.3671E-01</td>
<td>5.3634E-02</td>
<td>1.0765E-01</td>
<td>1.6385E-13</td>
</tr>
<tr>
<td>2</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>tot</td>
<td>4.0223E+00</td>
<td>8.3671E-01</td>
<td>5.3634E-02</td>
<td>1.0765E-01</td>
<td>1.6385E-13</td>
</tr>
</tbody>
</table>

### Source Calculation Information...

- Number of histories, source calculation: 250000
- CPU time required: 4.070522 sec
- Avg. no. of calls/Int.: 4.39
- Avg. source particle weight: 0.9980
- Min source particle weight: 0.9799
- Max source particle weight: 0.9998
- Weight balance: 1.6385E-13
- Particles generated through weight splitting: 0
- Weight gain = 0.0000E+00
- Weight loss = 0.0000E+00
- The maximum bank number was 0
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

...iteration controls and criteria...

***iteration criteria***

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>initer</td>
<td>inner iteration count until near lambda (i.e. transition source)</td>
<td>30</td>
<td>terminates inners</td>
</tr>
<tr>
<td>itli</td>
<td>inner iteration count when near lambda (i.e. transition source)</td>
<td>30</td>
<td>terminates inners</td>
</tr>
<tr>
<td>epti</td>
<td>fractional phase flux change per inner</td>
<td>1.0E-03</td>
<td>does another inner</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>outdr</td>
<td>sub-inner iteration count</td>
<td>40</td>
<td>terminates sub-outers</td>
</tr>
<tr>
<td>epti</td>
<td>diffusion lambda-1.0 (see note below)</td>
<td>1.0E-03</td>
<td>does another sub-inner</td>
</tr>
</tbody>
</table>

Note: epti when the problem is finally converged, will equal epti the value shown above. However, early in the iteration process, a larger value may be used to avoid unnecessary iterations.

Final convergence criteria

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>outir</td>
<td>outer iteration count</td>
<td>20</td>
<td>quits with error message</td>
</tr>
<tr>
<td>epti</td>
<td>transport lambda-1.0</td>
<td>1.0E-03</td>
<td>does another outer</td>
</tr>
</tbody>
</table>

...flux and eigenvalue convergence as monitored by twodant...

*Key start iteration monitor*

**1.45 0 9 0.01 0.001 1 9 0.01**

--- inner iteration summary for outer iteration no. 1 ---

1 per sec flux at group mesh change
1 9 0.01 0.001 25 23

**1.45 0 9 0.01 0.001 1 9 0.01**

--- flux change max pside ok max pside max pside inners
1.00000000 -1.00000000 5.64421E-04 0.00000000 yes

SSSSSS all convergence criteria satisfied SSSSSS

particle balance = 0.00000000 total inners all outers = 9
...Sample Problem 2 Output Listing...

the current fit coefficients are:

- number of histories, simo 1 link calculation = 83750
- avg. no. of col/i hist = 3.94
- min source particle weight = 0.9633
- max source particle weight = 0.9992
- weight balance = -6.661E-16

- particles generated through weight splitting = 0
- weight gain = 0.000000
- weight loss = 0.000000
- 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 flux to an source is 5.62E-01 g = 2 k = 280

...flux and eigenvalue convergence as monitored by twodant...

```plaintext
Start iteration monitor
```

```
- inner iteration summary for outer iteration no. 1 --

  iter per max flux at
group group change mesh
  1  8     0.335E+03  22, 40

  cpu time outer diffusion max phismax phis inner
  (sec) no. inner sub-outer source multip lambda-1 flux change flux change converged
  52.92  0     0.00000000  1.00000000  0.00000000  0.00000000  0.00000000  yes

  SSSSS all convergence criteria satisfied SSSSS
  particle balance = 0.00000000  total inner all outer = 8
```

...timo 1 link monitor...
...simo 3 link monitor...

the fit error in the incoming boundary flux is 3.15E-02    g = 1    i = 38
the fit error in the an to mc source is 0.066E-06    g = 0    k = 0
the current fit coefficients are:
5.358E-02  4.661E-03  2.812E-04

number of histories, simo 3 link calculation = 6000
cpu time required = 7.933E+00 sec
avg. no. of cells/evact = 3.98
avg. source particle weight = 0.1375
min source particle weight = 0.0016
max source particle weight = 0.6210
weight balance = 5.1076-15

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 an 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 (sec) no. inner sub-cutsers source multip lambda-1 max phase max phase inner flux change flux change converged
$53.45  0

inner iteration summary for outer iteration no. 1 —

iter per sec flux at
1  6  0.68E-03  25  40

*cpu time outer diffusion (sec) no. inner sub-cutsers source multip lambda-1 flux change flux change converged
$53.25  1  6  0  1.00000000 -1.00000000  6.789E-04  0.000000E+00  yes

$SS$ all convergence criteria satisfied $SS$  
particle balance = 0.000000E+00  total inner = all cutsers = 6

8
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

--- Sample Input 4 Link Monitor ---

...4 link monitor...

the fit error in the incoming boundary flux is 5.926E-03  g = 1  i = 40
the fit error in the an to re source is 0.00886-00  g = 0  k = 0
the current fit coefficients are:
5.534E-02  5.569E-03  5.315E-04  3.152E-05
number of histories, 4 link calculation = 6000
CPU time required = 7.817E+00 sec
avg. no. of coll/hist = 3.91
avg. source particle weight = 0.0125
sum source particle weight = 0.0001
max source particle weight = 0.0944
weight balance = 6.661E-15

particles generated through weight splitting = 0
weight gain = 0.0005-00
weight loss = 0.0005-00
the maximum bank number was 0
the iteration error in the outgoing boundary flux is 4.658E-02  g = 1  i = 38
the iteration error in the re to an source is 6.692E-02  g = 2  k = 280

--- Flux and Eigenvalue Convergence as Monitored by twodant ---

May start iteration monitor

CPU time outer diffusion max phi max phi max phi max phi inners
(sec) no. inners sub-outers source multip lambda-1 flux change flux change converged

--- inner iteration summary for outer iteration no. 1 ---

iter per max flux at

group group change max
1  5  0.768E-03  22  40

CPU time outer diffusion max phi max phi max phi max phi inners
(sec) no. inners sub-outers source multip lambda-1 flux change flux change converged
567.00 1  5  0 1.000000000 1.000000000 7.555322E-04 0.000086-00 yes

SSSSSS all convergence criteria satisfied $SSSSS$
particle balance = 0.000000000 total inners all outers = 5

TWODANT USER'S GUIDE Version 3.0 3-117
APPENDIX A: SAMPLE INPUT
Sample Problem 2: Output Listing

*****************************************************************************
...simno 5 link monitor...
*****************************************************************************

- the diffusion flux error in the incoming boundary flux is 8.82E-04, g = 1, i = 47
- the diffusion flux error in the sn to mc source is 0.003E-00, g = 0, k = 0
- the current fit coefficients are:
  5.803E-02, 5.990E-03, 6.890E-04, 6.618E-05
- the projected fit coefficients are:
  5.806E-02, 6.338E-03, 6.589E-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 sn to mc 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 phdse max phdse inners
  - (sec) no. iters sub-outers source multip lambda-1 flux change flux change converged
  - 567.22 0

- inner iteration summary for outer iteration no. 1

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>0.70E-03</td>
<td>22, 40</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>0.85E-03</td>
<td>6, 39</td>
<td></td>
</tr>
</tbody>
</table>

- *cpu time outer diffusion*  max phdse max phdse inners
  - (sec) no. iters sub-outers source multip lambda-1 flux change flux change converged
  - 573.13 1 14 0 1.00000000 -1.00000000 8.50250E-04 0.00000000 yes

$\text{*******}$ all convergence criteria satisfied $\text{*******}$
- particle balance = 1.27098E-05  total iters all outer = 14

*****************************************************************************
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

******************************************************************************
* .snm 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  0.003E+00  g = 0  k = 0
* the current fit coefficients are:
*  5.599E-02  6.258E-03  8.989E-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  2.98E-09  g = 2  k = 245
******************************************************************************

... monte carlo region particle balance ...

**GO sources sn sources fission arc net leakage absorption
1  1.000e+00  0.000e+00  0.000e+00  7.5904E-01  8.0318E-02
2  0.000e+00  0.000e+00  0.000e+00  0.000e+00  0.000e+00
   tot  1.000e+00  0.000e+00  0.000e+00  7.5904E-01  8.0318E-02
**GO in scatter self scatter out scatter particle balance
1  0.000e+00  6.0237e+00  1.6061E-01  3.0391E-05
2  1.5238E-17  0.000e+00  0.000e+00  0.000e+00
   tot  1.5238E-17  6.0237e+00  1.6061E-01  3.0391E-05

mc coarse mesh leakages

top leakages
  i 1.0615e+02  1.2998e-03  0.000e+00
   0.0169  0.0452  0.0000
bottom leakages
  i 1.3181e-01  9.8729e-02  0.000e+00
   0.0049  0.0556  0.0000
right leakages
  i 1.0477e+00
   0.0000
left leakages
  i 0.0000e+00
   0.0000

tracks entering (coarse mesh, in calc)

   i  1  5157  1.52284

******************************************************************************
Sample Problem 2: Output Listing

---

...system balance tables... (neutrons only)

Key start balance table

<table>
<thead>
<tr>
<th>Source</th>
<th>Fission Source</th>
<th>In Scattering</th>
<th>Self Scattering</th>
<th>Cut Scattering</th>
<th>Net Leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.000000E+00</td>
<td>0.000000E+00</td>
<td>-2.220446E+16</td>
<td>1.638145E+01</td>
<td>4.163571E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>4.163571E+01</td>
<td>2.199755E+00</td>
<td>2.220446E+01</td>
</tr>
<tr>
<td>Tot</td>
<td>1.000000E+00</td>
<td>0.000000E+00</td>
<td>4.163571E+01</td>
<td>1.858119E+01</td>
<td>4.163571E+01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Absorption</th>
<th>Particle Balance</th>
<th>Right Leakage</th>
<th>Horizontal Leakage</th>
<th>Top Leakage</th>
<th>Vertical Leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.081923E-03</td>
<td>1.895495E-05</td>
<td>9.499340E-03</td>
<td>9.499340E-03</td>
<td>2.044384E-02</td>
</tr>
<tr>
<td>2</td>
<td>3.968251E-01</td>
<td>-1.312661E-07</td>
<td>8.546637E-04</td>
<td>8.546637E-04</td>
<td>8.676684E-04</td>
</tr>
<tr>
<td>Tot</td>
<td>5.950317E-01</td>
<td>1.270885E-05</td>
<td>1.035400E-02</td>
<td>1.035400E-02</td>
<td>2.131150E-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Left Leakage</th>
<th>Bottom Leakage</th>
<th>Neutron Spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000E+00</td>
<td>3.454923E+01</td>
</tr>
<tr>
<td>2</td>
<td>0.000000E+00</td>
<td>2.379236E+02</td>
</tr>
<tr>
<td>Tot</td>
<td>0.000000E+00</td>
<td>3.692847E+01</td>
</tr>
</tbody>
</table>

Timing for sweep and UDA...tmax= 13.933 and t tolerance= 6.367 and tgray= 0.0000 seconds.

---

Integral Summary Information

Summary Eigenvalue Not Used

<table>
<thead>
<tr>
<th>Integral</th>
<th>Source</th>
<th>Neutron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integral</td>
<td>Source</td>
<td>1.000000E+00</td>
</tr>
<tr>
<td>Integral</td>
<td>Fission</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>Integral</td>
<td>In Scattering</td>
<td>4.163571E+01</td>
</tr>
<tr>
<td>Integral</td>
<td>Self Scattering</td>
<td>1.858119E+01</td>
</tr>
<tr>
<td>Integral</td>
<td>Cut Scattering</td>
<td>4.163571E+01</td>
</tr>
<tr>
<td>Integral</td>
<td>Absorption</td>
<td>5.990317E+01</td>
</tr>
<tr>
<td>Integral</td>
<td>Right Leakage</td>
<td>1.035400E+02</td>
</tr>
<tr>
<td>Integral</td>
<td>Horizontal Leakage</td>
<td>1.035400E+02</td>
</tr>
<tr>
<td>Integral</td>
<td>Top Leakage</td>
<td>2.131150E-02</td>
</tr>
<tr>
<td>Integral</td>
<td>Vertical Leakage</td>
<td>3.905940E-01</td>
</tr>
</tbody>
</table>
APPENDIX A: SAMPLE INPUT

Sample Problem 2: Output Listing

******************************************************************************
* key start coarse mesh balances *
******************************************************************************

absc + cutout + a(j- (left) + a(j+ (right)) + a(j- (bottom) + a(j+ (top)

q(fixed) + q(fission) + incut + a(j- (left) + a(j+ (right)) + a(j- (bottom) + a(j+ (top)

noted: boundary sources are included in the currents

mesh volumetric fission in- out- effective left j+ left j- bottom j+ bottom j- particle

| (i,j) fixed source source scatter scatter absorption times area times area times area times area times area balance |
|-------------------------------------------------|----------|----------------|----------------|----------------|----------------|----------------|----------------|--------------------|
| 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 |
| 2 0.0000E+00 0.0000E+00 1.0606E-01 8.3126E-02 8.3126E-02 1.7460E-02 8.4946E-03 0.0000E+00 9.6172E-02 |
| 3 0.0000E+00 0.0000E+00 2.5576E-01 1.2786E-01 1.2786E-01 1.0477E+00 5.4431E-01 0.0000E+00 1.0869E-01 |
| 4 9.4938E-03 3.8543E-18 5.4720E-04 0.0000E+00 0.0000E+00 1.4673E-01 2.6938E-12 |
| 5 1.2613E-02 0.0000E+00 2.9890E-03 0.0000E+00 1.8841E-03 0.0000E+00 |
| 6 0.0000E+00 0.0000E+00 0.0000E+00 4.1636E-01 2.0819E-01 2.0819E-01 0.0000E+00 0.0000E+00 1.8053E-05 |
| 7 5.4993E-03 3.8543E-18 2.1312E-02 0.0000E+00 0.0000E+00 1.3000E-05 |

...interface file reflux written...

...interface file cores written...

tweak: iteration time, mins 9.56118E-00
**APPENDIX A: SAMPLE INPUT**

Sample Problem 2: Output Listing

---

**run highlights**

*may start run highlights*

- all modules are tentatively go.
- interface file goed written.
- copy sections free cards.
- interface mixing files written.
- interface file agt written.
- m files each, goed written.
- interface file solving written.
- edit module execution suppressed.
- neither edit file nor edit cards exist.
- only 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.
- all convergence criteria met.
- interface file reflux written.
- interface file merge written.

---

**storage and timing history**

<table>
<thead>
<tr>
<th>module</th>
<th>cas</th>
<th>len</th>
<th>lay</th>
<th>cpu</th>
<th>smpn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>571.7</td>
</tr>
<tr>
<td>100</td>
<td>11036</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>2.0</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
</tr>
<tr>
<td>102</td>
<td>1041</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>103</td>
<td>39</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>2032</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>107</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>108</td>
<td>603</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>109</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>112</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>16090</td>
<td>40000</td>
<td>68478</td>
<td>140000</td>
<td>571.7</td>
</tr>
<tr>
<td>201</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>203</td>
<td>1204</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>204</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>205</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>408.0</td>
</tr>
<tr>
<td>206</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>22.4</td>
</tr>
<tr>
<td>207</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>208</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>209</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>210</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>211</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

...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

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

\(^a\) 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.
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

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

This work was supported by the US Department of Energy.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
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

## TABLE OF CONTENTS
- LIST OF FIGURES .................................................. 4-7
- LIST OF TABLES ..................................................... 4-9
- INTRODUCTION ....................................................... 4-11
- DOCUMENTATION FOR TWODANT/GQ USAGE ......................... 4-13
  - What Is In This User's Guide .................................... 4-13
  - What Is Available Elsewhere .................................... 4-14
- GEOMETRY CONCEPTS .................................................. 4-17
  - Submeshes ......................................................... 4-18
  - Objects ........................................................... 4-19
  - Components ....................................................... 4-20
  - Geometry .......................................................... 4-20
  - Boundary Conditions ............................................ 4-21
- TWODANT/GQ INPUT OVERVIEW ........................................ 4-23
  - Input Block Order ................................................ 4-23
  - Free Field Input Summary ....................................... 4-25
    - Arrays ................................................................ 4-25
    - Numeric Data Items ............................................ 4-25
    - Character Data Items ......................................... 4-25
    - Blocks ................................................................ 4-26
    - Strings ................................................................ 4-26
    - Comments .......................................................... 4-26
    - Operators .......................................................... 4-26
    - Frequently Used Operators .................................... 4-27
  - MINI-MANUAL Introduction ........................................ 4-28
  - MINI MANUAL ......................................................... 4-29
- TWODANT/GQ INPUT DETAILS ........................................... 4-33
  - Introduction ....................................................... 4-33
  - Title Line Details ................................................ 4-36
    - Title Line Control ............................................... 4-36
    - Block-I Details: Dimensions and Controls .................... 4-37
      - Dimensions ...................................................... 4-37
      - Storage Requirements ......................................... 4-38
      - Run Configuration Controls .................................. 4-38
    - Block-II Details: Geometry ..................................... 4-39
      - Spatial Resolution ............................................... 4-39
      - Submesh Definitions ........................................... 4-40
      - Zoning Definitions ............................................ 4-41
      - Object Definitions ............................................ 4-41
      - Component Definitions ....................................... 4-42
      - Geometry Definition ........................................... 4-43
      - Boundary Conditions .......................................... 4-44
      - Long Name Aliases ............................................. 4-45
    - Block-III Details: Nuclear Data ................................ 4-46

**TWODANT/GQ USER’S GUIDE**  Version 3.0  4-5
TABLE OF CONTENTS

Nuclear Data Type and Options ......................................................... 4-46
Alternate Library Name ................................................................. 4-48
Text Cross-Section Library Format ................................................ 4-50
Block-IV Details: Cross-Section Mixing ........................................... 4-52
MATLS input array .............................................................................. 4-53
Primary Mixing Arrays ........................................................................ 4-53
ASSIGN input array ............................................................................ 4-54
PREMIX input array ............................................................................ 4-54
Character Names vs. Numeric Names ................................................ 4-55
Mixing Array for a Concentration Search .......................................... 4-55
ASGMOD input array ........................................................................... 4-56
Concentration Modifier ....................................................................... 4-56
Miscellaneous Mixing Input .............................................................. 4-58
Block-V Details: Solver Input ............................................................. 4-59
Desired Calculation ............................................................................ 4-59
Iteration Controls ................................................................................ 4-60
Output Controls .................................................................................. 4-60
Miscellaneous Solver Input ............................................................... 4-61
Quadrature Details ............................................................................. 4-62
Flux Start ............................................................................................ 4-63
General Eigenvalue Search Control .................................................... 4-63
Dimension Search Input ..................................................................... 4-64
Concentration Search Input ............................................................... 4-64
Volumetric Source Options ............................................................... 4-65
Boundary Source Input ....................................................................... 4-67
Block-VI Details: Edit Input ................................................................. 4-68
Edit Spatial Specifications .................................................................. 4-68
Reaction Rates from Cross Sections .................................................. 4-69
Edit Cross-Section Types by Position and Name ................................. 4-70
Reaction Rates from User Response Functions .................................. 4-71
Energy Group Collapse Specifications .............................................. 4-72
Reaction Rate Summing ...................................................................... 4-73
Mass Inventories .................................................................................. 4-73
Power Normalization .......................................................................... 4-74
Miscellaneous Edit Items .................................................................... 4-75
Special Plot Linkage ........................................................................... 4-76
MENDF Library Edit Cross Sections .................................................. 4-77

REFERENCES ....................................................................................... 4-79

APPENDIX A: SAMPLE INPUT ............................................................... 4-81
Sample Problem: Supercell k_eff Calculation ...................................... 4-81
Sample Problem: Output Description ................................................ 4-82
Sample Problem: Output Listing ....................................................... 4-85

APPENDIX B: OPERATING SYSTEM SPECIFICS .................................. 4-95
UNIX/UNICOS Execution .................................................................. 4-95
Library Search Path ............................................................................ 4-96
LIST OF FIGURES

Figure 4.1: Submesh Examples ................................................................. 4-18
Figure 4.2: Examples of Objects ............................................................... 4-19
Figure 4.3: A Component Formed from Four Objects ............................ 4-20
Figure 4.4: The Final Geometry ............................................................... 4-21
Figure 4.5: Example of a Boundary Segment ........................................ 4-22
Figure 4.6: TWODANT/GQ Input Order ............................................... 4-24
Figure 4.7: Hypothetical Supercell Geometry ........................................... 4-81
LIST OF TABLES

Table 4.1: LIBNAME Availability ................................................................. 4-48
Table 4.2: UNIX Search Path................................................................. 4-96
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 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 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 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$_{\text{eff}}$ 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,

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.
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 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.
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 vertices. 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 vertices 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 differentzonings 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](image)

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.
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.
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:

\[ \text{SOURCX}= 0 0 0 1.1 1.1 0 0 0 0 0 \]

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:

\[ \text{SOURCX} = 0 \, 0 \, 0 \, 2R \, 1.1 \, F0 \]

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

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

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 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 P0 source values for each x mesh point in the first y mesh. The next string is the P0 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 P1 source values for each x mesh are given. After all P1 values are given, the P2 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
(316 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
NOEDITT
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

<table>
<thead>
<tr>
<th>LIB valid:</th>
<th>ODNINP</th>
<th>XSLIB</th>
<th>ISOTXS</th>
<th>GRUPXS</th>
<th>BXSLIB</th>
<th>MACRXS</th>
<th>MACBCD</th>
<th>XSLIBB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(local)MENDF</td>
<td>(local)MENDFG</td>
<td>alternate XSLIB name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| WRITMXS valid: | MACBCD | XSLIBB | XSLIBF | XSLIBE |

<table>
<thead>
<tr>
<th>LNG</th>
<th>BALXS</th>
<th>NTICHI</th>
<th>CHIVEC [NGROUP]</th>
<th>LIBNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of this block is needed only for text libraries.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| T |

### Block-IV: Mixing

<table>
<thead>
<tr>
<th>MATLS [-;MT]</th>
<th>ASSIGN [-;NZONE]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREMIX [;-]</td>
<td></td>
</tr>
<tr>
<td>ASGMOD [;-]</td>
<td></td>
</tr>
<tr>
<td>CMOD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MATNAM [MT]</th>
<th>ZONNAM [NZONE]</th>
<th>MATSPEC [-] valid: ATFRAC</th>
<th>WTFRAC</th>
<th>ATDEN</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ATWT [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
</tr>
</tbody>
</table>

### Block-V: Solver

<table>
<thead>
<tr>
<th>IEVT</th>
<th>ISCT</th>
<th>ITH</th>
<th>IBL</th>
<th>IBR</th>
<th>IBT</th>
<th>IBB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EPSI</th>
<th>ITTL</th>
<th>IIIM</th>
<th>OITM</th>
<th>ITLIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**iff LIB=ODNINP, insert ASCII text cross sections here**
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 -
SOURCE [IT;JT*NGROUP*NMQ]
- or -
SOURCE [IT;JT*NGROUP*NMQ] and SOURCF [IT;JT*NGROUP*NMQ]
- or -
SOURCE [NGROUP;NMQ] and SOURCF [IT;JT*NGROUP*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]

Block-VI: EDIT

PTED
ZNED

POINTS [K], K≤IT*JT
EDZONE [IT;JT]

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

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

ICOLL [K], K≤NGROUP
IGRPED

MICSUM [ -]
IRSUM [ -]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

PRPLTED

T
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
(formatted 3l6)
{Required}

<table>
<thead>
<tr>
<th>Word</th>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHEAD</td>
<td>Number of title lines that follow[^a]</td>
</tr>
<tr>
<td>2</td>
<td>NOTTY</td>
<td>Suppress output to on-line user terminal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0/1 = no/yes</td>
</tr>
<tr>
<td>3</td>
<td>NOLIST</td>
<td>Suppress listing of all ASCII text input?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0/1 = no/yes (default=no)</td>
</tr>
</tbody>
</table>

[^a]: Follow this control line with NHEAD title lines containing descriptive comments. Format is 12A6.
Block-I Details: Dimensions and Controls

Dimensions
{Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGEOM</td>
<td>\textit{Geometry}: 106/107 = generalized x-y/r-z or use one of the following character strings: \textit{X-Y-GQ, R-Z-GQ}</td>
</tr>
<tr>
<td>NGROUP</td>
<td>Number of energy groups.</td>
</tr>
<tr>
<td>ISN</td>
<td>\textit{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.)</td>
</tr>
<tr>
<td>NISO</td>
<td>Number of physical isotopes on the basic input cross-section library.</td>
</tr>
<tr>
<td>MT</td>
<td>Number of physical materials\textsuperscript{a} to be created.</td>
</tr>
<tr>
<td>NZONE</td>
<td>Number of geometric zones\textsuperscript{b} in problem.</td>
</tr>
<tr>
<td>IT</td>
<td>Number of mesh intervals in the x (or r) direction in the largest submesh.</td>
</tr>
<tr>
<td>JT</td>
<td>Number of mesh intervals in the y (or z) direction in the largest submesh.</td>
</tr>
</tbody>
</table>

\textsuperscript{a} Material is defined on page 4-53.

\textsuperscript{b} Zone is defined on page 7-13.
Storage Requirements
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSCM</td>
<td>Length of SCM desired (default=40000)</td>
</tr>
<tr>
<td>MAXLCM</td>
<td>Length of LCM desired (default=14000)</td>
</tr>
</tbody>
</table>

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}

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

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XYEPS</td>
<td>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.</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XVERT [-;N^a]</td>
<td>List of the x coordinates of all the vertices in each submesh.</td>
</tr>
<tr>
<td>YVERT [-;N]</td>
<td>List of the y coordinates of all the vertices in each submesh. Must be the same number and order as in XVERT.</td>
</tr>
<tr>
<td>SMNAME [1;N]</td>
<td>Name (up to 8 characters) of each submesh.</td>
</tr>
<tr>
<td></td>
<td>(default names = submsh1, submsh2, etc.)</td>
</tr>
<tr>
<td>SMIT [1;N]</td>
<td>Row length (in number of intervals) for each submesh.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>SMINTS [-;N]</td>
<td>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.</td>
</tr>
<tr>
<td>SMNCORS [-;N]</td>
<td>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).</td>
</tr>
</tbody>
</table>

---

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMZONES [M;NSMZ]</td>
<td>Zone number for each of the $M_n$ intervals in a submesh. $^g$ 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.</td>
</tr>
<tr>
<td>SMZNAME [1;NSMZ]</td>
<td>Name (up to 8 characters) of each zoning. (default names = zoning1, zoning2, etc.)</td>
</tr>
</tbody>
</table>

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$^a$}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTS [3;NOBJS]</td>
<td>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.</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPONS [6a:NC]</td>
<td>For each of the NC components, enter the following string:</td>
</tr>
<tr>
<td></td>
<td>[\text{objectname } x \ y \ \theta \ x_t \ y_t ;]</td>
</tr>
<tr>
<td></td>
<td>where ((x,y)) are the coordinates of a center of rotation in the object named \text{objectname}. (\theta) 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.)</td>
</tr>
<tr>
<td></td>
<td>If you wish to name the component explicitly, precede the objectname with the component name as in:</td>
</tr>
<tr>
<td></td>
<td>[\text{componentname objectname } x \ y \ \theta \ x_t \ y_t ;]</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>[\text{componentname } name_1 \ x_1 \ y_1 \ \theta_1 \ x_{i1} \ y_{i1} \</td>
</tr>
<tr>
<td></td>
<td>\text{name}<em>2 \ x_2 \ y_2 \ \theta_2 \ x</em>{i2} \ y_{i2} \ name_3 \ x_3 \ y_3 \ \theta_3 \</td>
</tr>
<tr>
<td></td>
<td>\text{name}_i \ x_i \ y_i \ \text{***} ;]</td>
</tr>
<tr>
<td></td>
<td>where (-\text{name}_i) may be either an object name or a component name and \text{***} signifies possible continuation of the pattern.</td>
</tr>
</tbody>
</table>

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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEOMETRY [5]</td>
<td>For the geometry enter the following sequence:</td>
</tr>
<tr>
<td></td>
<td>( name \ x \ y \ \theta \ x_t \ y_t ; )</td>
</tr>
<tr>
<td></td>
<td>where - ((x,y)) are the coordinates of a center of rotation in the object or component named ( name ). ( \theta ) 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.</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>( name_1 \ x_1 \ y_1 \ \theta_1 \ x_{t1} \ y_{t1} \ name_2 \ x_2 \ y_2 \ \theta_2 \ x_{t2} \ y_{t2} \ name_3 \ x_3 \ y_3 \ \theta_3 \ x_{t3} \ y_{t3} \ldots ; )</td>
</tr>
<tr>
<td></td>
<td>where - ( name_i ) may be either an object name or a component name and ( \ldots ) signifies possible continuation of the pattern.</td>
</tr>
</tbody>
</table>

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\textsuperscript{a}}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| BDRYSEG [5\textsuperscript{b};NBD] | For each of the NBD boundary segments\textsuperscript{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.  

\textsuperscript{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.  

\textsuperscript{b} The string length will be 6 if \( ib=7 \) or \( ib=8 \).  

\textsuperscript{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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFNAME</td>
<td>Long name for submeshes</td>
</tr>
<tr>
<td>SMFZNAME</td>
<td>Long name for submesh zonings</td>
</tr>
<tr>
<td>OBNAME</td>
<td>Long name for objects</td>
</tr>
<tr>
<td>CMFNAME</td>
<td>Long name for components</td>
</tr>
<tr>
<td>GEFNAME</td>
<td>Long name for the geometry</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIB</strong></td>
<td>Name(^a) and form of the cross-section data file. Enter as a data item one of the following words:</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><em>ISOTXS</em>(^b)</td>
<td>CCCC standard isotope ordered binary cross-section file.</td>
</tr>
<tr>
<td><em>XSLIB</em></td>
<td>ASCII text library supplied in a separate file named XSLIB.</td>
</tr>
<tr>
<td><em>ODNINP</em></td>
<td>ASCII text library follows after this block of input (after the T of Block-III).</td>
</tr>
<tr>
<td><em>GRUPXS</em>(^c)</td>
<td>CCCC standard group ordered cross-section file.</td>
</tr>
<tr>
<td><em>BXSLIB</em></td>
<td>Binary library supplied as a separate file named BXSLIB. [See “Binary Form of Card-Image Libraries (the BXSLIB file)” on page 10-12.</td>
</tr>
<tr>
<td><em>MACRXS</em>(^d)</td>
<td>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.</td>
</tr>
<tr>
<td><em>XSLIBB</em></td>
<td>See “XSLIBB Card-Image Library File” on page 10-12.</td>
</tr>
<tr>
<td><em>MACBCD</em></td>
<td>ASCII form of MACRXS file.</td>
</tr>
<tr>
<td><em>MENDF</em></td>
<td>(LANL only) See “The Los Alamos MENDF5 Cross-Section Library” on page 10-13.</td>
</tr>
<tr>
<td><em>MENDFG</em></td>
<td>(LANL only) See “The Los Alamos MENDF5G Gamma Cross-Section Library” on page 10-14.</td>
</tr>
<tr>
<td><strong>other</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

| WRITMXS \*optional* | Controls the code's writing certain ASCII cross-section files. Enter one of the following words: |
Nuclear Data Type and Options (Cont.)

{Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
<td></td>
</tr>
<tr>
<td>MACBCD</td>
<td>Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file.</td>
<td></td>
</tr>
<tr>
<td>XSLIBB</td>
<td>Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file.</td>
<td></td>
</tr>
<tr>
<td>XSLIBE</td>
<td>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).</td>
<td></td>
</tr>
<tr>
<td>XSLIBF</td>
<td>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).</td>
<td></td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>balance cross sections by adjusting absorption cross section.</td>
</tr>
<tr>
<td>0</td>
<td>do not balance cross sections. (default)</td>
</tr>
<tr>
<td>1</td>
<td>balance cross sections by adjusting self-scattering cross section.</td>
</tr>
</tbody>
</table>

**NTICHI** {optional} MENDF fission fraction to be used for the problem (LANL only). 

\[ \frac{1}{2}/3 = \text{Pu239/235/238} \] (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 JL1P1=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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBNAME</td>
<td>Alternate name of the library file. May be used only with certain types of libraries. See Table 4.1.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>LIB</th>
<th>LIBNAME AVAILABLE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>No</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>Yes</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>Yes</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>ODNINP</td>
<td>No</td>
</tr>
<tr>
<td>MACBCD</td>
<td>No</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>No</td>
</tr>
<tr>
<td>MENDFa</td>
<td>No</td>
</tr>
<tr>
<td>MENDFGb</td>
<td>No</td>
</tr>
<tr>
<td>XSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>other</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

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}

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

\[ \ldots \sigma_{\text{abs}}, \nu \sigma_f, \sigma_{\text{total}}, \ldots, \sigma_{g+2 \rightarrow g}, \sigma_{g+1 \rightarrow g}, \sigma_{g \rightarrow g}, \sigma_{g-1 \rightarrow g}, \sigma_{g-2 \rightarrow g}, \ldots \]

Notice that all terms in the scattering matrix are in positions relative to that of the selfscattering 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 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 $\Sigma_g$ 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 $\Sigma_g$ 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLS\textsuperscript{a} [-;MT]</td>
<td>Instructions for mixing &quot;isotopes&quot; or premixes into &quot;materials.” See details below.</td>
</tr>
<tr>
<td>ASSIGN\textsuperscript{b} [-;NZONE]</td>
<td>Assignments of materials to geometric zones. See below.</td>
</tr>
<tr>
<td>PREMIX [-;-]</td>
<td>Instructions for mixing &quot;isotopes&quot; into premixes. See below. {optional}</td>
</tr>
</tbody>
</table>

\textsuperscript{a} The information entered in the MATLS= array is written to the CCCC standard interface files NDX-SRF and ZNATDN.

\textsuperscript{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:

\[
\text{MATLS} = \text{mat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \ldots \text{etc}.
\]

where \( \text{mat}_1 \) is the desired character name of the first material and \( \text{comp}_1, \text{comp}_2 \), and so on are the character names of its components which have "densities" of, respectively, \( \text{den}_1, \text{den}_2 \), and so on. Additional materials (i.e., \( \text{mat}_2, \text{mat}_3 \), and so on up to the required number, \( \text{MT} \)) 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 \textit{den} 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:

\[
\text{MATLS= mat}_1 \text{ comp}_1; \quad \text{mat}_2 \text{ comp}_2 \text{ den}_2; \quad \text{etc.} \quad \text{etc.};
\]

\textbf{ASSIGN input array}

The general form of the ASSIGN instruction is shown below:

\[
\text{ASSIGN= zone}_1 \text{ mat}_1 \text{ vol}_1, \text{ mat}_2 \text{ vol}_2, \text{ etc.} \quad \text{etc.};
\]

where \textit{zone}_1 is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). \textit{mat}_1, \textit{mat}_2, and so on are the character names of the materials that will be present in this zone with, respectively, the “volume fractions” \textit{vol}_1, \textit{vol}_2, and so on. Additional zones (i.e., \textit{zone}_2, \textit{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.

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

\textbf{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:

\[
\text{PREMIX= tmat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \text{ etc.} \quad \text{etc.};
\]
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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASGMOD(^a)</td>
<td>([-; -]) ( C_1 ) parameters used in concentration searches. See the discussion below.</td>
</tr>
</tbody>
</table>

\( a. \) 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:

\[ C(z,x) = C_0(z,x) + C_1(z,x) \times \text{CMOD} \]

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:

\[ \text{ASGMOD} = \text{zone mat}_m \ \text{vol}_m, \ \text{mat}_n \ \text{vol}_n, \ ... \ \text{etc.} \ ... ; \]

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOD</td>
<td>Concentration modifier. Input value is not used in a search. See the discussion below.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATNAM [MT]</td>
<td>Character material names for Materials. Used only if the mat 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.</td>
</tr>
<tr>
<td>ZONNAM [NZONE]</td>
<td>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.</td>
</tr>
<tr>
<td>MATSPEC [≤MT]</td>
<td>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 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.</td>
</tr>
<tr>
<td>ATWT [≤2*NISO] (required b)</td>
<td>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= iso1 atwt1 iso2 atwt2 .... 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].</td>
</tr>
</tbody>
</table>

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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| IEVT | Calculation type: Enter one of the following values:.

<table>
<thead>
<tr>
<th>Value</th>
<th>Calculation Desired</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>inhomogeneous source (default)</td>
</tr>
<tr>
<td>1</td>
<td>$k_{\text{eff}}$</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha$ (time absorption) search</td>
</tr>
<tr>
<td>3</td>
<td>concentration search</td>
</tr>
<tr>
<td>4</td>
<td>dimension search</td>
</tr>
<tr>
<td>ISCT</td>
<td>Legendre order of scattering (default=0).</td>
</tr>
<tr>
<td>ITH</td>
<td>0/1 = direct/adjoint calculation (default=0).</td>
</tr>
<tr>
<td>IBL$^a$</td>
<td>Left boundary condition. 0/1/3 = vacuum/reflective/white.</td>
</tr>
<tr>
<td>IBR</td>
<td>Right boundary condition. 0/1/3 = vacuum/reflective/white.</td>
</tr>
<tr>
<td>IBT</td>
<td>Top boundary condition. 0/1/2/3 = vacuum/reflective/periodic/white.</td>
</tr>
<tr>
<td>IBB</td>
<td>Bottom boundary condition. 0/1/2/3 = vacuum/reflective/periodic/white.</td>
</tr>
</tbody>
</table>

---

*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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPSI</td>
<td>Convergence precision (default = 0.0001).</td>
</tr>
<tr>
<td>IITL</td>
<td>Maximum number of inner iterations per group at first (default = 1).</td>
</tr>
<tr>
<td>IITM</td>
<td>Maximum number of inners allowed when near fission source convergence (default chosen by code).</td>
</tr>
<tr>
<td>OITM</td>
<td>Maximum no. of outer iterations (default = 20).</td>
</tr>
</tbody>
</table>

### Output Controls
**{All Optional}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUXP</td>
<td>Final flux print. 0/1/2 = no/isotropic/all moments.</td>
</tr>
<tr>
<td>XSECTP</td>
<td>Cross-section print. 0/1/2 = no/principal/all.</td>
</tr>
<tr>
<td>FISSRP</td>
<td>Fission source rate print. 0/1 = no/yes.</td>
</tr>
<tr>
<td>SOURCP</td>
<td>Source print. 0/1/2/3 = no/as input/normalized/both.</td>
</tr>
<tr>
<td>ANGP</td>
<td>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.</td>
</tr>
<tr>
<td>RAFLUX</td>
<td>Prepare angular flux file (RAFLUX or AAFLUX). 0/1 = no/yes.</td>
</tr>
<tr>
<td>RMFLUX</td>
<td>Prepare flux moments file (RMFLUX or AMFLUX). 0/1 = no/yes.</td>
</tr>
</tbody>
</table>
## Miscellaneous Solver Input

### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TRCOR</strong></td>
<td>Apply transport correction(^a) to cross sections on MACRXS file. Enter one of the following words:</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>DIA</td>
<td>Use diagonal transport correction</td>
</tr>
<tr>
<td>BHS</td>
<td>Use Bell-Hansen-Sandmeier correction</td>
</tr>
<tr>
<td>CESARO</td>
<td>Use Cesaro “correction”</td>
</tr>
<tr>
<td>NO</td>
<td>(or omit entry) don’t apply correction</td>
</tr>
</tbody>
</table>

**NORM**

Normalize the fission source rate to this value when IEVT\(\geq 1\) 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 \(k_{\text{eff}}\) problems where the integral is equal to NORM*\(k_{\text{eff}}\)) 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 \(\text{cm}^{-1}\). 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]**

Density factor to use at each fine mesh point.

or

**DENX [IT]**\(^c\)

Density factor to use at each fine x-mesh (or r-mesh) (default=1).

and/or

**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 \(\text{DEN}(i,j,k)\), at mesh interval \((i,j,k)\) is computed as follows:

\[
\text{DEN}(i,j) = \text{DENX}(i)\times\text{DENY}(j)
\]
# Quadrature Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRPSN [NGROUP]</td>
<td>$S_n$ order to be used for each group.</td>
</tr>
<tr>
<td>IQUAD</td>
<td>Source of quadrature constants. Enter one of the following:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-3</td>
<td>Get constants from SNCONS file.</td>
</tr>
<tr>
<td>-2</td>
<td>Triangular Chebychev-Legendre built-in set. Any even value for ISN can be</td>
</tr>
<tr>
<td></td>
<td>used up to 50. From 50 to 100, ISN must be in multiples of 10. See Ref. 6</td>
</tr>
<tr>
<td></td>
<td>for details.</td>
</tr>
<tr>
<td>1</td>
<td>Traditional built-in constants. Any even value for ISN can be used between</td>
</tr>
<tr>
<td></td>
<td>2 and 16, inclusive. (This is the default).</td>
</tr>
<tr>
<td>2</td>
<td>Rectangular Chebychev-Legendre built-in set. (REQUIRES ISN negative in</td>
</tr>
<tr>
<td></td>
<td>Block-I) Any even value for the absolute value of ISN can be used up to 50.</td>
</tr>
<tr>
<td></td>
<td>From 50 to 100, the absolute value of ISN must be in multiples of 10. See</td>
</tr>
<tr>
<td></td>
<td>Ref. 6 for details.</td>
</tr>
<tr>
<td>WGT(^a) [MM(^b)]</td>
<td>Quadrature weights.</td>
</tr>
<tr>
<td>{optional}</td>
<td></td>
</tr>
<tr>
<td>MU [MM]</td>
<td>Mu cosines.</td>
</tr>
<tr>
<td>{optional}</td>
<td></td>
</tr>
<tr>
<td>ETA [MM]</td>
<td>Eta cosines.</td>
</tr>
<tr>
<td>{optional}</td>
<td></td>
</tr>
</tbody>
</table>

a. Presence of the WGT, MU, and ETA arrays overrides the IQUAD input.
b. $MM = ISN^2(ISN+2)/8$ for $iquad = -2, 1$.$MM = (ISN/2)^2$ for $iquad = 2$. 
Flux Start  
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFLUX</td>
<td>0/1 = no/yes Read flux start from the RTFLUX file.³</td>
</tr>
</tbody>
</table>

³. There is presently no ASCII text input flux guess available for TWODANT/GQ.

General Eigenvalue Search Controla  
{ IEVT >1}  

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

¹. 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*

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XM [IM]</td>
<td>x-dimension fractional change per mesh interval (or r).</td>
</tr>
<tr>
<td>YM [JM]</td>
<td>y-dimension fractional change per mesh interval (or z).</td>
</tr>
</tbody>
</table>

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 \(k+1\):

\[
X_{\text{VERT}}(i+1,j)^{k+1} = X_{\text{VERT}}(i,j)^{k} + \{X_{\text{VERT}}(i+1,j)-X_{\text{VERT}}(i,j)\} \times (1.0+EV*XM(i)) \quad i = 1, 2, ..., IM, \ j = 1, 2, ..., JM
\]

\[
Y_{\text{VERT}}(i+1,j)^{k+1} = Y_{\text{VERT}}(i,j)^{k} + \{Y_{\text{VERT}}(i+1,j)-Y_{\text{VERT}}(i,j)\} \times (1.0+EV*YM(i)) \quad i = 1, 2, ..., IM, \ j = 1, 2, ..., JM
\]

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 vertices 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*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

\(^a\) 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSORS</td>
<td>Read source from interface file FIXSRC. 0/1 = no/yes</td>
</tr>
<tr>
<td>SOURCE [NGROUP; NMQ]</td>
<td>Source spectrum for each of NMQ(^a) moments. (Spatial distribution is assumed to be flat with value unity)</td>
</tr>
<tr>
<td>SOURCX [IT;NMQ]</td>
<td>x (or r) spatial distribution for each moment.</td>
</tr>
<tr>
<td>SOURCY [JT;NMQ]</td>
<td>y (or z) spatial distribution for each moment.</td>
</tr>
<tr>
<td>SOURCE [NGROUP; NMQ]</td>
<td>Source spectrum.</td>
</tr>
<tr>
<td>SOURCX [IT;NMQ]</td>
<td>x (or r) spatial distribution for each moment.</td>
</tr>
<tr>
<td>SOURCY [JT;NMQ]</td>
<td>y (or z) spatial distribution for each moment.</td>
</tr>
<tr>
<td>SOURCF [IT;JT<em>NGROUP</em>NMQ]</td>
<td>Spatial distribution for each row, group, and moment.</td>
</tr>
<tr>
<td>SOURCE [NGROUP; NMQ]</td>
<td>Source spectrum.</td>
</tr>
<tr>
<td>SOURCF [IT; JT*NMQ]</td>
<td>Spatial distribution for each row and moment.</td>
</tr>
</tbody>
</table>

---

\(a\). 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:

\[
SOURCF(i,j,g,m) = SOURCE(g,m) \times SOURCX(i,m) \times SOURCY(j,m)
\]
## Boundary Source Input

{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----</td>
<td>For a text-input source, choose one of the following options:</td>
</tr>
<tr>
<td>Option 1: Isotropic Boundary Source.</td>
<td></td>
</tr>
<tr>
<td>SILEFT [NGROUP;IT]</td>
<td>Isotropic source on the left side. (Spectrum at each y mesh interval) (or z).</td>
</tr>
<tr>
<td>SIRITE [NGROUP;IT]</td>
<td>Isotropic source on the right side.</td>
</tr>
<tr>
<td>SIBOTT [NGROUP;IT]</td>
<td>Isotropic source on the bottom side.</td>
</tr>
<tr>
<td>SITOP [NGROUP;IT]</td>
<td>Isotropic source on the top side.</td>
</tr>
</tbody>
</table>

| Option 2: Full Angular Boundary Source. |                                       |
| SALEFT [MM*2,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. |                                       |
| BSLFTG [NGROUP] | Spectrum on left side. |
| BSLFTY [IT] | Spatial distribution on left side. |
| BSLFTA [MM*2] | Angular distribution on left side. |
| BSRITG [NGROUP] | Spectrum on right side. |
| BSRITY [IT] | 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*

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTED</td>
<td>Do edits by fine mesh. 0/1 = no/yes.</td>
</tr>
<tr>
<td>ZNED</td>
<td>Do edits by zone. 0/1 = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below).</td>
</tr>
<tr>
<td>POINTS[ ≤JT*JT]</td>
<td>Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points).</td>
</tr>
<tr>
<td>EDZONE [IT;JT]</td>
<td>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).</td>
</tr>
</tbody>
</table>

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

### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDXS (\leq) NEDT (\text{required}^c)</td>
<td>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 &quot;Edit Cross-Section Types by Position and Name&quot; on page 4-70 or &quot;MENDF Library Edit Cross Sections&quot; 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.</td>
</tr>
<tr>
<td>RESDNT</td>
<td>Do edits using the resident macroscopic cross section at each point. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDISOS (\leq) NISO</td>
<td>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).</td>
</tr>
<tr>
<td>EDCONS (\leq) NISO</td>
<td>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).</td>
</tr>
<tr>
<td>EDMATS (\leq) SMT</td>
<td>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).</td>
</tr>
<tr>
<td>XDF(^d) [IT]</td>
<td>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).</td>
</tr>
<tr>
<td>YDF [JT]</td>
<td></td>
</tr>
</tbody>
</table>
## Edit Cross-Section Types by Position and Name

<table>
<thead>
<tr>
<th>CROSS-SECTION INPUT VIA ISOTXS or GRUPXS</th>
<th>CROSS-SECTION INPUT VIA ASCII TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>EDIT Position</strong></td>
</tr>
<tr>
<td>chi</td>
<td>1</td>
</tr>
<tr>
<td>nu-fission</td>
<td>2</td>
</tr>
<tr>
<td>total</td>
<td>3</td>
</tr>
<tr>
<td>absorption</td>
<td>4</td>
</tr>
<tr>
<td>(n,p)</td>
<td>5</td>
</tr>
<tr>
<td>(n,d)</td>
<td>6</td>
</tr>
<tr>
<td>(n,t)</td>
<td>7</td>
</tr>
<tr>
<td>(n,alpha)</td>
<td>8</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>9</td>
</tr>
<tr>
<td>(n,gamma)</td>
<td>10</td>
</tr>
<tr>
<td>fission</td>
<td>11</td>
</tr>
<tr>
<td>transport</td>
<td>12</td>
</tr>
</tbody>
</table>

---

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}**

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

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:

\[
RSFX(i,m) \times RSFY(j,m) \times RSFE(g,m)
\]
Energy Group Collapse Specifications
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOLL [NBG]</td>
<td>Edit energy group collapsing option:</td>
</tr>
<tr>
<td></td>
<td>Number of SOLVER energy groups in each EDIT broad group.</td>
</tr>
<tr>
<td></td>
<td>The NBG entries must sum to NGROUP.</td>
</tr>
<tr>
<td></td>
<td>(default = 1 energy group per EDIT broad group).</td>
</tr>
<tr>
<td>IGRPED</td>
<td>Print option on energy groups. Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>0</td>
<td>Print energy group totals only</td>
</tr>
<tr>
<td>1</td>
<td>Print broad groups only</td>
</tr>
<tr>
<td>2</td>
<td>Print broad groups only (same as 1)</td>
</tr>
<tr>
<td>3</td>
<td>Print both broad groups and totals</td>
</tr>
</tbody>
</table>
Reaction Rate Summing  
{Optional} 

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICSUM [&lt;500 sums]</td>
<td>Cross-section reaction rate summing specifications.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>IRSUMS [&lt;500 sums]</td>
<td>Response function reaction rate summing specifications.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

Mass Inventories  
{Optional} 

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSED</td>
<td>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.</td>
</tr>
</tbody>
</table>
Power Normalization
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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

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

\(^a\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| PRPLTED | Write an ASCII file of the pointwise reaction rates to link to the TECPLOT\textsuperscript{\textcopyright} 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\textsuperscript{\textcopyright} 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\textsuperscript{\textcopyright} preprocessor.

If option 0 (print only) is chosen, no TECPLOT\textsuperscript{\textcopyright} 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

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi$</td>
<td>CHI</td>
<td>fission spectrum</td>
</tr>
<tr>
<td>$\nu \sigma_f$</td>
<td>NUSIGF</td>
<td>effective nu-sigma-fission</td>
</tr>
<tr>
<td>$\sigma_t$</td>
<td>TOTAL</td>
<td>Total cross section</td>
</tr>
<tr>
<td>$\sigma_a$</td>
<td>ABS</td>
<td>absorption$^a$</td>
</tr>
<tr>
<td>(n,n)</td>
<td>MEND1</td>
<td>elastic scattering</td>
</tr>
<tr>
<td>(n,n')</td>
<td>MEND2</td>
<td>inelastic scattering</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>MEND3</td>
<td>n,2n scattering</td>
</tr>
<tr>
<td>(n,3n)</td>
<td>MEND4</td>
<td>n,3n scattering</td>
</tr>
<tr>
<td>(n,\gamma)</td>
<td>MEND5</td>
<td>gamma production</td>
</tr>
<tr>
<td>(n,\alpha)</td>
<td>MEND6</td>
<td>alpha production</td>
</tr>
<tr>
<td>(n,p)</td>
<td>MEND7</td>
<td>proton production</td>
</tr>
<tr>
<td>(n,f)</td>
<td>MEND8</td>
<td>direct fission</td>
</tr>
<tr>
<td>(n,n')f</td>
<td>MEND9</td>
<td>second-chance fission</td>
</tr>
<tr>
<td>(n,2n)f</td>
<td>MEND10</td>
<td>third-chance fission</td>
</tr>
<tr>
<td>(n,F)</td>
<td>N-FISS</td>
<td>$[(n,F) = (n,f) + (n,n')f + (n,2n)f]$</td>
</tr>
<tr>
<td>$\chi_p$</td>
<td>MEND12</td>
<td>prompt fission spectrum</td>
</tr>
<tr>
<td>$\chi_t$</td>
<td>MEND13</td>
<td>total fission spectrum</td>
</tr>
</tbody>
</table>

$^a \sigma_a$ for group $g$ is defined as $\sigma_a = \sigma_t - \sum_{g' \rightarrow g} \sigma_{g' \rightarrow g}$

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


Sample Problem: Supercell $k_{\text{eff}}$ 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, CTTRLTUBE, 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 outscater 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 02/17/95 with Solver Version 01-04-95Beta --- Release 3.0 Machine e

---

### Listing of cards in the input stream

<table>
<thead>
<tr>
<th>Card</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>twodant/fg test case</td>
</tr>
<tr>
<td>4</td>
<td>Hypothetical seven assembly supercell</td>
</tr>
<tr>
<td>5</td>
<td>/ the complete geometry is a set of 7 hex assemblies</td>
</tr>
<tr>
<td>6</td>
<td>/ there are two types of assemblies</td>
</tr>
<tr>
<td>7</td>
<td>/ 1. control assembly</td>
</tr>
<tr>
<td>8</td>
<td>/ 2. fuel assembly with seven hexagonal subassemblies</td>
</tr>
<tr>
<td>11</td>
<td>/ and represent an infinite array of these supercells repeating</td>
</tr>
<tr>
<td>13</td>
<td>/ along a roughly northwest-southeast line</td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>

### Block I - Controls and Dimensions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
<tr>
<td>&quot;&lt;&quot;</td>
<td>4</td>
</tr>
</tbody>
</table>

---

### Block II - Geometry

---

### Special Resolution

---

### Submesh Ramps

---

### Row Length for Each Submesh

---

### Block III - Cross sections

---

## TWODANT/GQ User's Guide

---

Version 3.0

---
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>94</td>
<td>0.5</td>
<td>0.57</td>
<td>1.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>cladding material</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>fluid surrounding the control rod</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.9</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>polon mixture inside the control rod</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.9</td>
<td>0.0</td>
<td>1.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

 bonard: ---

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>/</td>
<td>***** Block-IV - Mixing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>material</td>
<td></td>
<td>each isotope is a mat</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>assign</td>
<td></td>
<td>mats in each zone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>coolant</td>
<td>cool</td>
<td>pure fuel coolant</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>homogen fuel</td>
<td>0.95</td>
<td>cool 0.03 clad 0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>control clad</td>
<td>0.05</td>
<td>polon 0.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>control fluid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>control clad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>control clad</td>
<td>t</td>
<td>control coolant</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>/</td>
<td>***** Block-V - Solution Desired</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>key</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>end</td>
<td>0</td>
<td></td>
<td>k-eff, p-zero calculation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>k-eff</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>p-zero</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>/</td>
<td>***** Block-VI - No Bakes Desired</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

4-86 TWODANT/GQ USER'S GUIDE Version 3.0
---

**APPENDIX A: SAMPLE INPUT**

---

**Sample Problem: Output Listing**

---

```plaintext
...case title...

---

**key start case input**

---

**output/gq test case**

---

**hypothetical seven assembly supercall**

---

```plaintext
...block i - controls and dimensions...

---

```plaintext
...dimensions...

---

```plaintext
106 igcon 6/7/9/11/106/107 x-y-z/triangles/r-theta/gq x-y/gq x-z
1 ngroup number of energy groups
2 lam angular quadrature order
5 nile number of input isotopes (from isotopes, groups, or cards)
5 st number of parent materials
6 mone 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...
readin= 30000
record= 89000
---

```plaintext
...key end block ii read-gcon

---

```plaintext
...key end block iii read-es...

---

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

---

```plaintext
...library source...
...libcard...
...card library parameters (array name = cards)...

---

```plaintext
0 record medium legendre order to be found in input cross sections
5 stl lower position in cross section table
5 fh position of total cross section
4 ih position of self scatter cross sections
0 ifdo -1/0/1/2 = off/(ke18.6)/off/linear fido/free fido library
1 jtit 0/1 - no/yes there is a title card before each table
0 ifzpl 0/1 - no/yes library higher order scattering contains 2-1 factor
0 subes 0/1 - no/yes save binary xalib file (fidoexe=xalib)
1 awkord 0/1 - full fido read/quick fido read (default=quick)

---

```plaintext
...energy structure...

---

```plaintext
  group   chi    vel   lower bound   upper bound   group   chi    vel   lower bound   upper bound
1  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00

---

```plaintext
last neutron group (ing) is number 1

---

```plaintext
0 halos -1/0/1 - adjust absorption/no/adjust self scatter to force 2 balance

---

```plaintext
...edit position names...

---

```plaintext
card
position abase position
1 chi
2 rnsdif  2
3 total  3
4 abs  1

---

```

---

**TWODANT/GQ USER'S GUIDE**

Version 3.0

4-87
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Listing

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

<table>
<thead>
<tr>
<th>isotope</th>
<th>isotope name</th>
<th>order</th>
<th>header card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>cool</td>
<td>p0</td>
<td>- coolant throughout the supercell</td>
</tr>
<tr>
<td>2.</td>
<td>fuel</td>
<td>p0</td>
<td>- fuel mixture inside fuel rods</td>
</tr>
<tr>
<td>3.</td>
<td>clad</td>
<td>p0</td>
<td>- cladting material</td>
</tr>
<tr>
<td>4.</td>
<td>fluid</td>
<td>p0</td>
<td>- fluid surrounding the control rod</td>
</tr>
<tr>
<td>5.</td>
<td>poison</td>
<td>p0</td>
<td>- poison mixture inside the control rod</td>
</tr>
</tbody>
</table>

...mixing instructions...

<table>
<thead>
<tr>
<th>mix</th>
<th>comp</th>
<th>density</th>
<th>comp</th>
<th>density etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. cool</td>
<td>cool</td>
<td>1.0000E+00,</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>2. fuel</td>
<td>fuel</td>
<td>1.0000E+00,</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>3. clad</td>
<td>clad</td>
<td>1.0000E+00,</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>4. fluid</td>
<td>fluid</td>
<td>1.0000E+00,</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>5. poison</td>
<td>poison</td>
<td>1.0000E+00,</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

assign

| 1. coolant | cool         | 1.0000E+00, | -      |
| 2. backup fuel | fuel         | 9.5000E-01, 9.5000E-02, cool | 1.0000E+02, 2.0000E+02, clad |
| 3. control clad | clad         | 5.0000E-02, 9.5000E-01, cool | 1.0000E+00, |
| 4. stringgap fluid | fluid       | 1.0000E+00, | -      |
| 5. exchanger clad | clad         | 1.0000E+00, | -      |
| 6. control coolant | coolant     | 1.0000E+00, | -      |
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Listing

this twodant/gq problem run on 03/17/95 with solver version 02-16-95 beta --- release 3.0 machine e
hypothetical seven assembly supernode

************************************************************************************

... block v — solver input ...

raw as input defaulted

... required input (array name = solin) ...
1 1 int 0/1/2/3/4 - type of calculation
0 homogeneous source
1 k-effective
2 alpha or time absorption search
3 concentration search
4 delta (i.e. dimension) search
0 0 int legendre order of scattering
0 0 ich 0/1 - direct/adjoint - mode of calculation (default-direct)
0 0 int 0/1/3 - left boundary condition vacuum/reflective/white
0 0 int 0/1/2 - right boundary condition vacuum/reflective/white
0 0 int 0/1/2/3 - top boundary condition vacuum/reflective/white
0 0 int 0/1/2/3 - bottom boundary condition vacuum/reflective/periodic/white

... convergence controls (array name = iter) ...
1.000E-04 1.000E-04 epdi inner iteration convergence criterion (default=0.0001)
1 1 intl maximum number of inner iterations per group until fission source is near convergence, i.e. lethality is near convergence (default=1)
0 30 itlm maximum number of inner iterations per group when close to fission source convergence (default calculated)
100 100 itim maximum number of outer iterations (default=20)
0 0 rtim iteration time limit (seconds)

************************************************************************************

... block v — solver input (continued) ...

raw as input defaulted

... miscellaneous parameters (array name = misc) ...
0.000E+00 0.000E+00 bhtp backing height
1.000E+00 1.000E+00 norm normalization factor
0 0 influx 0/1/0/yes - read input flux from file dfux (after flux for adjoint)
0 0 influx 0/1/0/yes - read input source from file dsource
0 1 igead -1/2/3/4 - source of quadrature constants (default=1)
1 0 injep file
2 hybrid product set (triangular arrangement)
3 old bottom built-in set
2 product set (rectangular arrangement)
3 card input

... output controls (array name = solout) ...
0 0 fluxp 0/1/2 none/isotropic/all moments - flux print
0 0 specnp 0/1/2 none/principal/all - mesoscopic cross section print
0 0 efluxp 0/1/0/yes - print final fission source rate
0 0 ssource 0/1/2 none/read normalized/branch - print homogeneous source
0 0 angp 0/1/0/yes - print angular fluxes
0 0 mflux 0/1/0/yes - write angular fluxes to file dfuxes or dfuxes (if itb=1)
0 0 mflux 0/1/0/yes - write flux moments to file dfuxes
0 0 balp 0/1/0/yes - print coarse mesh balances

... parameters inferred from input arrays ...
 1 inchl 0/1/2 none/one chi/mono/multi chi
 0 izfrm 0/1/0 - none/x density vector/full matrix
 0 izdty 0/1/0 - use y density vector
 0 izsrc source symmetry
 0 iscm number of source moments input
 0 iscm number of source moments output
 0 iscm number of source moments input
 0 iscm number of source moments output
 0 iscm number of source moments input
 0 iscm number of source moments output
 0 igl -1/0/1/2 isotropic/cvoice/angles/vectors - left boundary source
 0 igl -1/0/1/2 isotropic/cvoice/angles/vectors - left boundary source
 0 igl -1/0/1/2 isotropic/cvoice/angles/vectors - top boundary source
 0 igl -1/0/1/2 isotropic/cvoice/angles/vectors - bottom boundary source

************************************************************************************
Sample Problem: Output Listing

---

**** parameters from block i...

106 igenm 106/107 x-y-z
1 ngroup number of energy groups
8 Ian angular quadrature order
5 nz number of material zones
24 in  number of fine mesh x intervals
24 ji number of fine mesh y intervals


---

***material assignments to zones...

* key start zones to zones...

<table>
<thead>
<tr>
<th>entry</th>
<th>zone name</th>
<th>material name</th>
<th>c0</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coolant</td>
<td>1 cool</td>
<td>1.000000E+00</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>homofuel</td>
<td>2 fuel</td>
<td>9.500000E-01</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>homofuel</td>
<td>1 cool</td>
<td>1.000000E-02</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>homofuel</td>
<td>3 clad</td>
<td>2.000000E-02</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>concrod</td>
<td>5 polon</td>
<td>5.000000E-02</td>
<td>0.000000E+00</td>
</tr>
<tr>
<td>6</td>
<td>core</td>
<td>6 cool</td>
<td>1.000000E+00</td>
<td>0.000000E+00</td>
</tr>
</tbody>
</table>

---

* key start storage map...

* key start storage map...

---

*** storage summary...

| total ncm required for this problem | 11037 |
| maximum ncm available (maxesm)   | 3096 |
| ncm required for transport        | 51594 |
| ncm required for diffusion        | 46554 |

---

*** storage summary...

| lcn selected for this problem | 11037 |
| maximum lcn specified (maxnom) | 3096 |
| storage required for all quantities in core is | 11037 |
| diffusion maxes are in core. If O.K. to put in core, set MAXOM up. | 11037 |
| flux maxes are in core. If O.K. to put in core, set MAXFL up. | 11037 |
| diffusion fluxes are in core. If O.K. to put in core, set MAXOM up. | 11037 |
| scalar tr fluxes are in core. If O.K. to put in core, set MAXOM up. | 11037 |
| storage required for all quantities on disk is | 11037 |

11037 words lcn required lcomah
* zero flux

---

*** key start an constants...

---

<table>
<thead>
<tr>
<th>an constants</th>
<th>nu</th>
<th>etas</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.192237476E+00</td>
<td>0.962299488E+00</td>
<td>0.291971170E-01</td>
</tr>
<tr>
<td>2</td>
<td>0.577352076E+00</td>
<td>0.793322178E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>3</td>
<td>0.192237476E+00</td>
<td>0.793322178E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>4</td>
<td>0.793322178E+00</td>
<td>0.577352076E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>5</td>
<td>0.577352076E+00</td>
<td>0.793322178E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>6</td>
<td>0.192237476E+00</td>
<td>0.577352076E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>7</td>
<td>0.793322178E+00</td>
<td>0.962299488E+00</td>
<td>0.291971170E-01</td>
</tr>
<tr>
<td>8</td>
<td>0.577352076E+00</td>
<td>0.192237476E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>9</td>
<td>0.192237476E+00</td>
<td>0.192237476E+00</td>
<td>0.233131808E-01</td>
</tr>
<tr>
<td>10</td>
<td>0.577352076E+00</td>
<td>0.192237476E+00</td>
<td>0.233131808E-01</td>
</tr>
</tbody>
</table>

---

4-90 TWODANT/GQ USER'S GUIDE  Version 3.0
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Listing

******************************************material map******************************************

material map for 3 mesh intervals 24, to 1 and 1 mesh intervals 1, to 24.

```
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 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

...cross section related data from file macro 03/17/05; version 1 ...

1 cool 2 fuel 3 clad 4 fluid 5 poison

...iteration controls and criteria...

***iteration criteria***

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>niit3 - inner iteration count until next lambda (i.e. fusion source convergence)</td>
<td>1</td>
<td>terminates inner</td>
<td></td>
</tr>
<tr>
<td>niit3 - inner iteration count when next lambda (i.e. fusion source convergence)</td>
<td>30</td>
<td>terminates inner</td>
<td></td>
</tr>
<tr>
<td>eppi - fractional power flux change per inner</td>
<td>1.00E-04</td>
<td>does another inner</td>
<td></td>
</tr>
</tbody>
</table>

**diffusion sub-criteria**

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>cim - sub-inner iteration count</td>
<td>200</td>
<td>terminates sub-inner</td>
<td></td>
</tr>
<tr>
<td>eppi - fractional power flux change per sub-inner (see note below)</td>
<td>1.00E-04</td>
<td>does another sub-inner</td>
<td></td>
</tr>
<tr>
<td>eppi - fractional power flux change per sub-inner (see note below)</td>
<td>1.00E-04</td>
<td>does another sub-inner</td>
<td></td>
</tr>
</tbody>
</table>

Notes eppi, when the problem is finally converged, will equal eppi, the value shown above. However, early in the iteration process, a larger value may be used to avoid unnecessary iterations.

**final convergence criteria**

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>cim - outer iteration count</td>
<td>100</td>
<td>quits with error message</td>
<td></td>
</tr>
<tr>
<td>eppi - transport lambda-1.0</td>
<td>1.00E-04</td>
<td>does another outer</td>
<td></td>
</tr>
</tbody>
</table>

TWODANT/GQ USER'S GUIDE Version 3.0
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Listing

*****************************************************************************************************************************************

...flux and eigenvalue convergence as monitored by sentinel...

*****************************************************************************************************************************************

May start iteration monitor *

*****************************************************************************************************************************************

<table>
<thead>
<tr>
<th>cpu time</th>
<th>outer diffusion</th>
<th>k-eff</th>
<th>max_phs</th>
<th>max_phs</th>
<th>inners</th>
</tr>
</thead>
<tbody>
<tr>
<td>(sec)</td>
<td>no.</td>
<td>sub-outers</td>
<td>eigenvalue</td>
<td>lambda-1</td>
<td>flux change</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>-----------</td>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>4.45</td>
<td>0</td>
<td>0</td>
<td>0.994139E5</td>
<td>4.74215E-03</td>
<td>0.00000002-00</td>
</tr>
<tr>
<td>4.15</td>
<td>1</td>
<td>0</td>
<td>1.00649E5</td>
<td>4.87315E-03</td>
<td>0.00000002-00</td>
</tr>
<tr>
<td>4.00</td>
<td>2</td>
<td>0</td>
<td>1.01081E5</td>
<td>4.92315E-03</td>
<td>0.00000002-00</td>
</tr>
<tr>
<td>4.80</td>
<td>3</td>
<td>0</td>
<td>1.01029E5</td>
<td>4.93315E-03</td>
<td>0.00000002-00</td>
</tr>
<tr>
<td>4.70</td>
<td>4</td>
<td>0</td>
<td>1.01029E5</td>
<td>4.93315E-03</td>
<td>0.00000002-00</td>
</tr>
<tr>
<td>4.60</td>
<td>5</td>
<td>0</td>
<td>1.01029E5</td>
<td>4.93315E-03</td>
<td>0.00000002-00</td>
</tr>
</tbody>
</table>

inner iteration summary for outer iteration 6

iter per max flux at group group change mesh
1 7 0.74E-04 22, 7

cpu time | diffusion | k-eff | max_phs | max_phs | inners |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(sec)</td>
<td>no.</td>
<td>sub-outers</td>
<td>eigenvalue</td>
<td>lambda-1</td>
<td>flux change</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>-----------</td>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>5.51</td>
<td>6</td>
<td>7</td>
<td>1.01029E5</td>
<td>4.93315E-03</td>
<td>0.00000002-00</td>
</tr>
</tbody>
</table>

$\text{SSS}SS$ all convergence criteria satisfied $\text{SSS}SS$ $\

particle balance = -1.92403E-06 total inners all outers = 15

*****************************************************************************************************************************************

*****************************************************************************************************************************************
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Listing

*****************************************************************************
* ...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
* gp  left leakage  bottom leakage  npdf spectrum
* 1  7.7169732E-05  -6.8418494E-03  1.0000000e+00
* tot  7.7169732E-05  -6.8418494E-03  1.0000000e+00
*****************************************************************************
* Timing data...tord, tswap, tdata =  0.6  0.8  0.6 CPU seconds.
*****************************************************************************
* integral summary information
* summary integral-N-eff   1.0197472e+00
* integral-source-i neutron  0.0000000e+00
* integral-fission-i neutron  1.0000000e+00
* integral-in-scatt-i neutron  0.0000000e+00
* integral-self-scatt-i neutron  3.5451763e+00
* integral-out-scatt-i neutron  1.4210855e-14
* integral-net-leakage-i neutron  2.0156399e-06
* integral-absorption-i neutron  9.999999E-01
* integral-right-leakage-i neutron  -7.6130496E-05
* integral-horizontal-leakage-i neutron  1.0366833E-05
* integral-top-leakage-i neutron  6.8428274E-03
* integral-vertical-leakage-i neutron  9.7895660E-07
* ...
* ...interface file rflux written..
* ...
* ...interface file ancore written..
*****************************************************************************
* twodant iteration time, mins  9.29185-02
*****************************************************************************
**APPENDIX A: SAMPLE INPUT**

---

**Sample Problem: Output Listing**

---

*run highlights*

- all modules are tentatively go.
- interface file gosamt written.
- copies sections from cards.
- interface mixing files written.
- interface file segment written.
- xe files memp written.
- interface file scaling written.
- edit module execution suppressed.
- neither edit nor edit cards exist.
- start solver execution.
- all convergence criteria met.
- interface file Reflex writing.
- interface file scores written.

---

**storage and timing history**

<table>
<thead>
<tr>
<th>module</th>
<th>words</th>
<th>limit</th>
<th>words</th>
<th>limit</th>
<th>seconds</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5.6</td>
<td>0.6</td>
</tr>
<tr>
<td>101</td>
<td>88845</td>
<td>89000</td>
<td>0</td>
<td>0</td>
<td>3.3</td>
<td>0.3</td>
</tr>
<tr>
<td>103</td>
<td>88854</td>
<td>89000</td>
<td>477</td>
<td>30000</td>
<td>0.2</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>2038</td>
<td>89000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>184</td>
<td>89000</td>
<td>35</td>
<td>30000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>108</td>
<td>606</td>
<td>89000</td>
<td>3</td>
<td>30000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>112</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>88847</td>
<td>89000</td>
<td>1037</td>
<td>30000</td>
<td>2.3</td>
<td>0.3</td>
</tr>
<tr>
<td>201</td>
<td>88847</td>
<td>89000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>202</td>
<td>4033</td>
<td>89000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>204</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>205</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>206</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>207</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>208</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>210</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>211</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

...execution terminated...

---

4-94

**TWODANT/GQ USER'S GUIDE**  Version 3.0
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:

\[
\text{dant.x} \ < \ odninp \ > \ odnout
\]

where \( \text{dant.x} \) is the name of the executable file, \( \text{odninp} \) is the user's choice for a name for the input file, and \( \text{odnout} \) is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is dant.x.

STDOUT contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDOUT 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**

<table>
<thead>
<tr>
<th>LIB</th>
<th>SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>Current Working Directory (CWD).</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>SNXSPATH, then CWD, but see text below.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>None, the library is contained in the input file.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>CWD</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>CWD</td>
</tr>
<tr>
<td>MENDF&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.</td>
</tr>
<tr>
<td>MENDFG&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>XSLIB</td>
<td>SNXSPATH, then CWD</td>
</tr>
<tr>
<td>other</td>
<td>For any name other than those above, the program will assume the form is XSLIB and search for it in SNXSPATH, then CWD.</td>
</tr>
</tbody>
</table>

<sup>a</sup> Available only at Los Alamos.

<sup>b</sup> 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.
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

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

This work was supported by the US Department of Energy.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
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** ................................................................. 5-7

**LIST OF TABLES** ............................................................... 5-9

**INTRODUCTION** ................................................................. 5-11

**DOCUMENTATION FOR TWOHEX USAGE** ......................... 5-13
   What Is In This User’s Guide ................................................ 5-13
   What Is Available Elsewhere ............................................... 5-14

**TWOHEX INPUT OVERVIEW** ............................................... 5-17
   Input Block Order ............................................................... 5-17
   Free Field Input Summary .................................................. 5-19
       Arrays ............................................................................... 5-19
       Numeric Data Items ....................................................... 5-19
       Character Data Items ..................................................... 5-19
       Blocks ............................................................................ 5-20
       Strings ........................................................................... 5-20
       Comments ...................................................................... 5-20
       Operators ....................................................................... 5-20
       Frequently Used Operators ............................................ 5-21
   MINI-MANUAL Introduction ............................................... 5-22
   MINI MANUAL ................................................................. 5-23

**TWOHEX INPUT DETAILS** ................................................... 5-27
   Introduction ........................................................................ 5-27
   Title Line Details .............................................................. 5-30
       Title Line Control ......................................................... 5-30
   Block-I Details: Dimensions and Controls ......................... 5-31
       Dimensions ................................................................... 5-31
       Storage Requirements ................................................. 5-32
       Run Configuration Controls ...................................... 5-32
   Block-II Details: Geometry ................................................ 5-33
       Geometry Arrays ........................................................... 5-33
   Block-III Details: Nuclear Data .......................................... 5-34
       Nuclear Data Type and Options .................................... 5-34
       Alternate Library Name ................................................. 5-36
       Text Cross-Section Library Format ............................... 5-38
   Block-IV Details: Cross-Section Mixing ............................. 5-40
       MATLS input array ........................................................ 5-41
       Primary Mixing Arrays .................................................. 5-41
       ASSIGN input array ....................................................... 5-42
       PREMIX input array ........................................................ 5-42
       Character Names vs. Numeric Names ......................... 5-43
       Concentration Search ..................................................... 5-43
       Miscellaneous Mixing Input .......................................... 5-44
   Block-V Details: Solver Input ............................................... 5-45
       Desired Calculation ....................................................... 5-45
       Iteration Controls ........................................................ 5-45

*MINI-MANUAL Introduction and MINI MANUAL are not included in the preview.*
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Controls</td>
<td>5-46</td>
</tr>
<tr>
<td>Miscellaneous Solver Input</td>
<td>5-46</td>
</tr>
<tr>
<td>Flux Guess From a File</td>
<td>5-47</td>
</tr>
<tr>
<td>Quadrature Details</td>
<td>5-47</td>
</tr>
<tr>
<td>Volumetric Source Options</td>
<td>5-49</td>
</tr>
<tr>
<td>Block-VI Details: Edit Input</td>
<td>5-50</td>
</tr>
<tr>
<td>Edit Spatial Specifications</td>
<td>5-50</td>
</tr>
<tr>
<td>Reaction Rates from Cross Sections</td>
<td>5-51</td>
</tr>
<tr>
<td>Edit Cross-Section Types by Position and Name</td>
<td>5-52</td>
</tr>
<tr>
<td>Reaction Rates from User Response Functions</td>
<td>5-53</td>
</tr>
<tr>
<td>Energy Group Collapse Specifications</td>
<td>5-54</td>
</tr>
<tr>
<td>Reaction Rate Summing</td>
<td>5-55</td>
</tr>
<tr>
<td>Mass Inventories</td>
<td>5-55</td>
</tr>
<tr>
<td>Power Normalization</td>
<td>5-56</td>
</tr>
<tr>
<td>Miscellaneous Edit Items</td>
<td>5-57</td>
</tr>
<tr>
<td>MENDF Library Edit Cross Sections</td>
<td>5-58</td>
</tr>
</tbody>
</table>

REFERENCES ........................................................................................................... 5-59

APPENDIX A: SAMPLE INPUT .............................................................................. 5-61
    Sample Problem: Standard $k_{eff}$ Calculation. ........................................ 5-61
    Sample Problem: Output Description .......................................................... 5-62

APPENDIX B: OPERATING SYSTEM SPECIFICS .............................................. 5-75
    UNIX/UNICOS Execution ............................................................................... 5-75
    Library Search Path ..................................................................................... 5-76
LIST OF FIGURES

Figure 5.1: TWOHEX Input Order ................................................................. 5-18
Figure 5.2: Possible $S_6$ Quadrature Arrangements ..................................... 5-48
Figure 5.3: Core Map of the Sample Problem .............................................. 5-61
Figure 5.4: Mesh Model for the Sample Problem ........................................ 5-61
LIST OF FIGURES
LIST OF TABLES

Table 5.1: LIBNAME Availability ................................................................. 5-36
Table 5.2: UNIX Search Path................................................................. 5-76
LIST OF TABLES
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,
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.
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.
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

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

---

\( ^a \) 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 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
(316 Format)
NHEAD,NOTTY,NOLIST

Title Line(s)

(IF NHEAD>0)

Block-I: Controls & Dimensions

IGEOM
NGROUP
IN
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]
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

Block-IV: Mixing

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

PREMIX [-;]

MATSPEC [-]
valid: ATFRAC
WTFRAC
ATDEN

ATWT [-]
MATNAM [MT]
ZONNAM [NZONE]
T
TWOHEX USER'S GUIDE

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-
DENV [IT], DENJ [JT]

--- Volumetric Source ---

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

T

Block-VI: EDIT

PTED
ZNED

POINTS [K], K<IT*JT
EDZONE [IT;JT]

EDXS [K], K<NEEDT
RESNNT
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
IGRPE

MCUM [-]
IRSUMS [-]

MASED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYYOLP
AJED
FLUXONE

T
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 316)a
{Required}

<table>
<thead>
<tr>
<th>Word</th>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHEAD</td>
<td>Number of title lines that follow. b</td>
</tr>
<tr>
<td>2</td>
<td>NOTTY</td>
<td>Suppress output to on-line user terminal?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0/1 = no/yes.</td>
</tr>
<tr>
<td>3</td>
<td>NOLIST</td>
<td>Suppress listing of all ASCII text input?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0/1 = no/yes. (default=no)</td>
</tr>
</tbody>
</table>

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.
### Dimensions {Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSCM</td>
<td>Length of SCM desired (default=40000_{10})</td>
</tr>
<tr>
<td>MAXLCM</td>
<td>Length of LCM desired (default=140000{10})</td>
</tr>
</tbody>
</table>

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}

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

Note: Default on all these controls is no.
# Geometry Arrays

{Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMAIN</td>
<td>Specifies the orientation and boundary conditions of the solution domain. Enter as a character data item one of the following five character words.</td>
</tr>
<tr>
<td>Word</td>
<td>Description</td>
</tr>
<tr>
<td>THIRD</td>
<td>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.</td>
</tr>
<tr>
<td>SIXTH</td>
<td>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.</td>
</tr>
<tr>
<td>WHOLE</td>
<td>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.</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>Specifies the height of a single triangle.</td>
</tr>
<tr>
<td>ZONES [IM;J?A]</td>
<td>Zone number 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.</td>
</tr>
<tr>
<td>BSQ [1]</td>
<td>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.</td>
</tr>
<tr>
<td>{optional}</td>
<td>CAUTION: Subsequent input of a buckling height via the BHGT array in the SOLVER block input will override the BSQ input.</td>
</tr>
</tbody>
</table>

---

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}*

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIB</strong></td>
<td>Name and form of the cross-section data file. Enter as a data item one of the following words:</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>ISO$^T_5$</td>
<td>CCCC standard isotope ordered binary cross-section file.</td>
</tr>
<tr>
<td>XSLIB</td>
<td>ASCII text library supplied in a separate file named XSLIB.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>ASCII text library follows after this block of input (after the T of Block-III).</td>
</tr>
<tr>
<td>GRUPX$^T_5$</td>
<td>CCCC standard group ordered cross-section file.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Binary library supplied as a separate file named BXSLIB. (See “Binary Form of Card-Image Libraries (the BXSLIB file)” on page 10-12.</td>
</tr>
<tr>
<td>MACRX$^S$</td>
<td>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.</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>See “XSLIBB Card-Image Library File” on page 10-12.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>ASCII form of MACRXS file.</td>
</tr>
<tr>
<td>MENDF</td>
<td>(LANL only) See “The Los Alamos MENDF5 Cross-Section Library” on page 10-13.</td>
</tr>
<tr>
<td>MENDFG</td>
<td>(LANL only) See “The Los Alamos MENDF5G Gamma Cross-Section Library” on page 10-14.</td>
</tr>
<tr>
<td>other</td>
<td>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.</td>
</tr>
<tr>
<td>WRITMXS</td>
<td>Controls the code’s writing certain ASCII cross-section files.</td>
</tr>
<tr>
<td>{optional}</td>
<td>Enter one of the following words:</td>
</tr>
</tbody>
</table>
Nuclear Data Type and Options (Cont.)
{Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Word</strong></td>
</tr>
<tr>
<td>MACBCD</td>
<td>Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file.</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file.</td>
</tr>
<tr>
<td>XSLIBE</td>
<td>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).</td>
</tr>
<tr>
<td>XSLIBF</td>
<td>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).</td>
</tr>
<tr>
<td>LNG</td>
<td>Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from gammas in the edits.</td>
</tr>
<tr>
<td>BALXS</td>
<td>Cross-section balance control. Enter one of the following values: WARNING See page 10-21 before using!</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NTICHI</td>
<td>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.</td>
</tr>
<tr>
<td>CHIVEC</td>
<td>Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.</td>
</tr>
</tbody>
</table>

---

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 12LP1=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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBNAME</td>
<td>Alternate name of the library file. May be used only with certain types of libraries. See Table 5.1.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>LIB</th>
<th>LIBNAME AVAILABLE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>No</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>Yes</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>Yes</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>ODNINP</td>
<td>No</td>
</tr>
<tr>
<td>MACBCD</td>
<td>No</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>No</td>
</tr>
<tr>
<td>MENDF&lt;sup&gt;a&lt;/sup&gt;</td>
<td>No</td>
</tr>
<tr>
<td>MENDFG&lt;sup&gt;b&lt;/sup&gt;</td>
<td>No</td>
</tr>
<tr>
<td>XSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>other</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

<sup>a</sup> Available only at Los Alamos.
<sup>b</sup> 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXORD</td>
<td>Highest Legendre order in the scattering tables.</td>
</tr>
<tr>
<td>IHM</td>
<td>Number of positions (entries) in each row of the cross-section table.</td>
</tr>
<tr>
<td>IHT</td>
<td>Position number of the total cross section.</td>
</tr>
<tr>
<td>IHS</td>
<td>Position number of the self-scatter cross section. (default = IHT + 1).</td>
</tr>
<tr>
<td>IFIDO</td>
<td>Format of the cross-section library. -1/0/1/2 = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.</td>
</tr>
<tr>
<td>ITITL</td>
<td>A title line precedes each table. 0/I = no/yes</td>
</tr>
<tr>
<td>IL/LP1</td>
<td>Higher order scattering cross sections on the library contain the 2L+1 term. 0/I = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.</td>
</tr>
<tr>
<td>SAVBXS</td>
<td>Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. 0/I = no/yes.</td>
</tr>
<tr>
<td>KWIKRD</td>
<td>Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? 0/I = no/yes (default=Yes).</td>
</tr>
<tr>
<td>NAMES [NISO]</td>
<td>Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . . . etc.).</td>
</tr>
<tr>
<td>EDNAME [IHT-3]</td>
<td>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.).</td>
</tr>
<tr>
<td>NTPI [NISO]</td>
<td>Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).</td>
</tr>
<tr>
<td>VEL [NGROUP]</td>
<td>Speeds for each group. Needed only for alpha calculations.</td>
</tr>
<tr>
<td>EBOUND [NGROUP+1]</td>
<td>Energy boundaries for each group.</td>
</tr>
</tbody>
</table>
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:

\[ \cdots \sigma_{\text{abs}}, \nu \sigma_f, \sigma_{\text{total}}, \cdots, \sigma_{g+2\rightarrow g}, \sigma_{g+1\rightarrow g}, \sigma_{g\rightarrow g}, \sigma_{g-1\rightarrow g}, \sigma_{g-2\rightarrow g}, \cdots \]

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 $\Sigma_g$ 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 $\Sigma_g$ 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLS&lt;sup&gt;a&lt;/sup&gt; [-;MT]</td>
<td>Instructions for mixing “isotopes” or premixes into “materials.” See details below.</td>
</tr>
<tr>
<td>ASSIGN&lt;sup&gt;b&lt;/sup&gt; [-;NZONE]</td>
<td>Assignments of materials to geometric zones. See below.</td>
</tr>
<tr>
<td>PREMIX [-;</td>
<td>-] {optional}</td>
</tr>
</tbody>
</table>

---

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:

```
MATLS= mat1 comp1 den1, comp2 den2, ...etc....;
```

where mat<sub>1</sub> is the desired character name of the first material and comp<sub>1</sub>, comp<sub>2</sub>, and so on are the character names of its components which have “densities” of, respectively, den<sub>1</sub>, den<sub>2</sub>, and so on. Additional materials (i.e., mat<sub>2</sub>, mat<sub>3</sub>, 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 \( den \) 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:

\[
\text{MATLS=} \text{mat}_1 \text{ comp}_1; \text{ mat}_2 \text{ comp}_2 \text{ den}_2; \ldots \text{etc.}.;
\]

**ASSIGN input array**

The general form of the ASSIGN instruction is shown below:

\[
\text{ASSIGN=} \text{zone}_1 \text{ mat}_1 \text{ vol}_1, \text{ mat}_2 \text{ vol}_2, \ldots \text{etc.}.;
\]

where \( \text{zone}_1 \) is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). \( \text{mat}_1, \text{mat}_2, \) and so on are the character names of the materials that will be present in this zone with, respectively, the “volume fractions” \( \text{vol}_1, \text{vol}_2, \) and so on. Additional zones (i.e., \( \text{zone}_2, \text{zone}_3, \) and so on up to the required number, \( \text{NZONE} \)) 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:

\[
\text{PREMIX=} \text{mat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \ldots \text{etc.}.;
\]
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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATNAM [MT]</td>
<td>Character material names for Materials. Used only if the mat1 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.</td>
</tr>
<tr>
<td>ZONNAM [NZONE]</td>
<td>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.</td>
</tr>
<tr>
<td>MATSPEC [≤MT]</td>
<td>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 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.</td>
</tr>
<tr>
<td>ATWT [≤2*NISO]</td>
<td>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= \textit{iso}_1 \textit{atwt}_1 \textit{iso}_2 \textit{atwt}_2 .... where \textit{iso}_n is the isotope name (identifier) for isotope n on the cross-section library and \textit{atwt}_n is that isotope's atomic weight. [See “Using Atomic Fractions or Weight Fractions (MATSPEC)” on page 11-13].</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEVT</td>
<td>Calculation type: Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>0</td>
<td>source</td>
</tr>
<tr>
<td>1</td>
<td>$k_{eff}^{a}$</td>
</tr>
<tr>
<td>ISCT</td>
<td>Legendre order of scattering.</td>
</tr>
<tr>
<td>ITH</td>
<td>0/1 = direct/adjoint calculation.</td>
</tr>
</tbody>
</table>

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPSI</td>
<td>Convergence precision (default=0.0001).</td>
</tr>
<tr>
<td>OITM</td>
<td>Maximum no. of outer iterations (default=20).</td>
</tr>
<tr>
<td>ITLIM</td>
<td>Number of seconds time limit (default=unlimited).</td>
</tr>
</tbody>
</table>
Output Controls
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUXP</td>
<td>Final flux print. 0/1/2 = no/isotropic/all moments.</td>
</tr>
<tr>
<td>XSECTP</td>
<td>Cross-section print. 0/1/2 = no/principal/all.</td>
</tr>
<tr>
<td>FISSRP</td>
<td>Fission rate print. 0/1= no/yes.</td>
</tr>
<tr>
<td>SOURCP</td>
<td>Source print. 0/1/2/3 = no/unnormalized/normalized/both.</td>
</tr>
</tbody>
</table>

Miscellaneous Solver Input
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORM</td>
<td>Normalize the fission source rate to this value when IEVT≥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.</td>
</tr>
<tr>
<td>BHGT</td>
<td>Buckling height. (in cm. if macroscopic cross section is in cm$^{-1}$.) (default=0.0 which is treated as infinity.)</td>
</tr>
<tr>
<td>CHI [NGROUP;M]</td>
<td>Fission fraction born into each group. Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.</td>
</tr>
<tr>
<td>DENX [IT]$^b$ and/or</td>
<td>Density factor to use for each x-mesh column (default=1). Applied to the zone macroscopic cross sections at each mesh interval.</td>
</tr>
<tr>
<td>DENY [JT]</td>
<td>Density factor to use for each y-mesh band (default=1).</td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| INFLUX  | Read the initial flux guess from a file. $0/1 = \text{no/yes.}$  
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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSORS</td>
<td>Read source from interface file FIXSRC. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>

----- 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:

**SOURCX [IT;NMQ]**\(^b\)  x spatial distribution for each moment.
**SOURCY [IT;NMQ]**  y spatial distribution for each moment.
(Spectrum is assumed to be flat with value unity)

Option 3:

**SOURCE [NGROUP; NMQ]**
Source spectrum.
**SOURCX [IT;NMQ]**  x spatial distribution for each moment.
**SOURCY [IT;NMQ]**  y spatial distribution for each moment.

Option 4:

**SOURCF [IT;JT*NGROUP*NMQ]**
Spatial distribution for each row, group, and moment.

Option 5:

**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 formulating 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:

\[
\text{SOURCF}(i,j,g,m) = \text{SOURCE}(g,m) \times \text{SOURCX}(i,m) \times \text{SOURCY}(j,m)
\]
Block-VI Details: Edit Input

Edit Spatial Specifications
{Required\textsuperscript{a}}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTED</td>
<td>Do edits by mesh interval. 0/1 = no/yes.</td>
</tr>
<tr>
<td>ZNED</td>
<td>Do edits by zone. 0/1 = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below).</td>
</tr>
<tr>
<td>POINTS[ ≤IT*JT] {optional}</td>
<td>Mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points)</td>
</tr>
<tr>
<td>EDZONE [IT;JT] {optional}</td>
<td>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).</td>
</tr>
</tbody>
</table>

\textsuperscript{a} Either PTED or ZNED or both must be unity in order to produce reaction rate edits.
## Reaction Rates from Cross Sections

*Optional*

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDXS [(\leq) NEDT] ({\text{required}\text{c}})</td>
<td>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.</td>
</tr>
<tr>
<td>RESDNT</td>
<td>Do edits using the resident macroscopic cross section at each point. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDISOS [(\leq) NISO]</td>
<td>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).</td>
</tr>
<tr>
<td>EDCONS [(\leq) NISO]</td>
<td>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).</td>
</tr>
<tr>
<td>EDMATS [(\leq) MT]</td>
<td>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).</td>
</tr>
<tr>
<td>XDF(^d) [IT]</td>
<td>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).</td>
</tr>
<tr>
<td>YDF [JT]</td>
<td></td>
</tr>
</tbody>
</table>

---

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:  
   \[ \text{XDF}(i) \times \text{YDF}(j) \]
## Edit Cross-Section Types by Position and Name

<table>
<thead>
<tr>
<th>CROSS-SECTION INPUT VIA ISOTXS or GRUPXS</th>
<th>CROSS-SECTION INPUT VIA ASCII TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>EDIT Position</td>
</tr>
<tr>
<td>chi</td>
<td>1</td>
</tr>
<tr>
<td>nu-fission</td>
<td>2</td>
</tr>
<tr>
<td>total</td>
<td>3</td>
</tr>
<tr>
<td>absorption</td>
<td>4</td>
</tr>
<tr>
<td>(n,p)</td>
<td>5</td>
</tr>
<tr>
<td>(n,d)</td>
<td>6</td>
</tr>
<tr>
<td>(n,t)</td>
<td>7</td>
</tr>
<tr>
<td>(n, alpha)</td>
<td>8</td>
</tr>
<tr>
<td>(n, 2n)</td>
<td>9</td>
</tr>
<tr>
<td>(n, gamma)</td>
<td>10</td>
</tr>
<tr>
<td>fission</td>
<td>11</td>
</tr>
<tr>
<td>transport</td>
<td>12</td>
</tr>
</tbody>
</table>

<sup>a</sup> Names are eight characters. A period within a name in this table denotes a blank.

<sup>b</sup> 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.

<sup>c</sup> 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\textsuperscript{a}}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSFE [NGROUP;M]</td>
<td>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.</td>
</tr>
<tr>
<td>RSFX [JT;M]\textsuperscript{b}</td>
<td>Response function x distribution for M functions.</td>
</tr>
<tr>
<td>RSFY [JT;M]</td>
<td>Response function y distribution for M functions.</td>
</tr>
<tr>
<td>{optional}</td>
<td>The above data are entered as M strings of IT or JT entries beginning with mesh point 1. (default=1.0)</td>
</tr>
<tr>
<td>RSFNM [M]</td>
<td>Character names for the user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM)</td>
</tr>
</tbody>
</table>

\textsuperscript{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.

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

\[ \text{RSFX}(i,m) \times \text{RSFY}(j,m) \times \text{RSFE}(g,m) \]
## Energy Group Collapse Specifications

### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICSUM [&lt;500 sums]</td>
<td>Cross-section reaction rate summing specifications.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
<tr>
<td>IRSUMS [&lt;500 sums]</td>
<td>Response function reaction rate summing specifications.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

Mass Inventories
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSED</td>
<td>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.</td>
</tr>
</tbody>
</table>
## Power Normalization

**{Optional}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>Normalize to POWER megawatts.(^a)</td>
</tr>
<tr>
<td>{required}</td>
<td>All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be</td>
</tr>
<tr>
<td></td>
<td>normalized. Fluxes are normally not printed here in the EDIT module, although they may be</td>
</tr>
<tr>
<td></td>
<td>extracted by using a unit response function. Any such fluxes will also be normalized to</td>
</tr>
<tr>
<td></td>
<td>POWER.</td>
</tr>
<tr>
<td></td>
<td>Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the</td>
</tr>
<tr>
<td></td>
<td>SOLVER Block (see NORM on page 5-45).</td>
</tr>
<tr>
<td>MEVPER</td>
<td>MeV released per fission (default=210 MeV). This value will be used along with the calculated</td>
</tr>
<tr>
<td>{required}</td>
<td>fission rate to determine the power.</td>
</tr>
<tr>
<td></td>
<td>For the power calculation, TWOHEX needs to know which cross section is the fission cross</td>
</tr>
<tr>
<td></td>
<td>section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or</td>
</tr>
<tr>
<td></td>
<td>GRUFXS library that designation is automatically provided (See “Edit Cross-Section Types by</td>
</tr>
<tr>
<td></td>
<td>Position and Name” on page 5-52). But if one uses an ASCII text library, either ODNINP or</td>
</tr>
<tr>
<td></td>
<td>XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 5-38).</td>
</tr>
</tbody>
</table>

\(^a\) Note that this normalization is meaningless if you are using the results of an adjoint run.
## Miscellaneous Edit Items
### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RZFLUX</td>
<td>Write the CCCC standard zone(^a) flux file RZFLUX or AZFLUX. 0/1 = no/yes.</td>
</tr>
<tr>
<td>RZMFLX</td>
<td>Write the code-dependent zone(^b) flux moments file RZMFLX or AZMFLUX. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDOUTF(^c)</td>
<td>ASCII output files control. Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-3</td>
<td>Write both EDTOGX (without scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>-2</td>
<td>Write EDTOGX file (without scalar fluxes).</td>
</tr>
<tr>
<td>0</td>
<td>Write neither file. (default)</td>
</tr>
<tr>
<td>1</td>
<td>Write EDTOUT file.</td>
</tr>
<tr>
<td>2</td>
<td>Write EDTOGX file (with scalar fluxes).</td>
</tr>
<tr>
<td>3</td>
<td>Write both EDTOGX (with scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>BYVOLP</td>
<td>Printed point reaction rates will have been multiplied by the mesh volume. 0/1 = no/yes.</td>
</tr>
<tr>
<td>AJED(^d)</td>
<td>Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. 0/1 = regular/adjoint.</td>
</tr>
<tr>
<td>FLUXONE</td>
<td>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.</td>
</tr>
</tbody>
</table>

---

\(^{a}\) 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**

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi$</td>
<td>CHI</td>
<td>fission spectrum</td>
</tr>
<tr>
<td>$\nu\sigma_f$</td>
<td>NUSIGF</td>
<td>effective nu-sigma-fission</td>
</tr>
<tr>
<td>$\sigma_t$</td>
<td>TOTAL</td>
<td>Total cross section</td>
</tr>
<tr>
<td>$\sigma_a$</td>
<td>ABS</td>
<td>absorption$^a$</td>
</tr>
<tr>
<td>(n,n)</td>
<td>MEND1</td>
<td>elastic scattering</td>
</tr>
<tr>
<td>(n,n')</td>
<td>MEND2</td>
<td>inelastic scattering</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>MEND3</td>
<td>n,2n scattering</td>
</tr>
<tr>
<td>(n,3n)</td>
<td>MEND4</td>
<td>n,3n scattering</td>
</tr>
<tr>
<td>(n,\gamma)</td>
<td>MEND5</td>
<td>gamma production</td>
</tr>
<tr>
<td>(n,\alpha)</td>
<td>MEND6</td>
<td>alpha production</td>
</tr>
<tr>
<td>(n,p)</td>
<td>MEND7</td>
<td>proton production</td>
</tr>
<tr>
<td>(n,f)</td>
<td>MEND8</td>
<td>direct fission</td>
</tr>
<tr>
<td>(n,n')f</td>
<td>MEND9</td>
<td>second-chance fission</td>
</tr>
<tr>
<td>(n,2n)f</td>
<td>MEND10</td>
<td>third-chance fission</td>
</tr>
<tr>
<td>(n,F)</td>
<td>N-FISS</td>
<td>$[(n,F) = (n,f) + (n,n')f + (n,2n)f]$</td>
</tr>
<tr>
<td>$\chi_p$</td>
<td>MEND12</td>
<td>prompt fission spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(only for fissionable materials)</td>
</tr>
<tr>
<td>$\chi_t$</td>
<td>MEND13</td>
<td>total fission spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(only for fissionable materials)</td>
</tr>
</tbody>
</table>

---

*a. $\sigma_a$ for group g is defined as $\sigma_a = \chi - \sum \sigma_{g \rightarrow g'}$.*

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


Sample Problem: Standard $k_{\text{eff}}$ 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.

\( \text{P}_0 \) 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.
Sample Problem: Output Description

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

1. 2 0 0
2. 2. SIMPLE PROBLEM FOR TWOHEX USER'S GUIDE
3. 3. SIMPLIFIED K CALCULATION. ALL INPUT BY MEANS OF ASCII TEXT
4. 5. \textbf{CROSS SECTIONS} - DICH OR WITH PERIPHERAL BURNING
5. 6. \textbf{MIXING} - DICHORIC DATA IN TEXT. LOS ALAMOS (UPF) FORMAT
6. 7. \textbf{SOLAR} - DEVICES NEEDED TO MAKE MATERIALS NAMED STEEL.
7. 8. \textbf{FUEL, AND SUBMARINE} - MATERIALS ASSEMBLED TO MAKE ZONES NAMED FUEL.
8. 9. \textbf{AND REFLECTOR} - \textbf{SOLAR} TEXT INPUT SUPPLIED.
9. 11. \textbf{EDITS} - \textbf{JOIN SETS} FOR RESIDENT MATERIALS
10. 12. \textbf{BLOCK} I
11. 13. \textbf{BLOCK} II (CROSS SECTIONS)
12. 14. \textbf{BLOCK} III (CROSS SECTIONS)
13. 15. \textbf{BLOCK} IV (MIXING)
14. 16. \textbf{BLOCK} V (EDITS)

---

\textbf{Notes:}
- **END OF CROSS-SECTION DATA ****
- ****\textbf{END OF BLOCK IV MIXING}****
- ****\textbf{END OF BLOCK V EDITS}****

---

\textbf{References:}
- \textbf{BIBLIOGRAPHY}
- \textbf{APPENDIX A: SAMPLE INPUT}
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Description

*****************************************************************************

case title
*****************************************************************************

*Key start case input.*

- 2 rhed number of title cards to follow
- 9 rotcy 0/1 no/yes suppress go-line terminal output
- 0 nolist 0/1 no/yes suppress input listing

*****************************************************************************

* SAMPLE PROBLEM FOR TWOHEX USER'S GUIDE
* STANDARD XL CALCULATION. ALL INPUT BY MEANS OF ASCII TEXT
*
*****************************************************************************

*Key end. block 1 read.*

*****************************************************************************

* dimensions *

  9 igem 6/7/8/9/11/106/107 x-y/z/triangles/z-theta/gq x-y/gq r-z
  2 ngroup number of energy groups
  4 ian angular quadrature order
  7 nuis number of input isotopes (from isotbs, groups, or cards)
  3 ml number of permanent materials
  2 nzone number of zones
  12 iu number of fine mesh intervals
  6 ji number of fine mesh y intervals

...storage...

  medim= 140000
  measim= 40000

*****************************************************************************

*Key end. block 2 read geom.*

*****************************************************************************

  9 igem 1/2/3/6/7/8/9/11/14/106/107

  soich domain calculational domain

  1.0000E25=01 height triangle height

*****************************************************************************

*Key end. block 3 read-x.*

*****************************************************************************
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Description

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

...library source...

libcross

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

0 record maximum legendre order to be found in input cross sections
1 imx last position in cross section table
2 imx position of total cross section
3 imx position of self scatter cross section
4 flfo -1/0/1/2 = diff/ea/fixed/linear fido free fido library
5 idnl 0/1 - no/yes there is a table card before this table
6 sifo 0/1 - no/yes library higher order scattering contains 2+1 factor
7 halx 0/1 - full fido read/quick fido read (default-quick)

...energy structure...

<table>
<thead>
<tr>
<th>Group</th>
<th>Chi</th>
<th>Vel</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Group</th>
<th>Chi</th>
<th>Vel</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>2</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
</tr>
</tbody>
</table>

last neutron group(ing) is number 2

0 halx -1/0/1 - adjust absorption/no/adjust self scatter to force ns balance

...edit position name...

card position where position

1 | chi
2 | mass
3 | total
4 | abs
5 | chid

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

<table>
<thead>
<tr>
<th>Isotope</th>
<th>Isotope Order</th>
<th>Header Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-16</td>
<td>O-16 (0-16)</td>
</tr>
<tr>
<td>2</td>
<td>1-16</td>
<td>(0-16)</td>
</tr>
<tr>
<td>3</td>
<td>2-16</td>
<td>(0-16)</td>
</tr>
<tr>
<td>4</td>
<td>3-16</td>
<td>(0-16)</td>
</tr>
<tr>
<td>5</td>
<td>4-16</td>
<td>(0-16)</td>
</tr>
<tr>
<td>6</td>
<td>5-16</td>
<td>(0-16)</td>
</tr>
<tr>
<td>7</td>
<td>6-16</td>
<td>(0-16)</td>
</tr>
</tbody>
</table>

...end card like read...

...end block iv read-data...

...end card like read...

...end block iv read-data...

...end card like read...

...end block iv read-data...

...end card like read...

...end block iv read-data...

...end card like read...

...end block iv read-data...

...end card like read...

...end block iv read-data...
... mixing instructions ...

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>density</th>
<th></th>
<th>density etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mix</td>
<td>comp</td>
<td></td>
<td>comp</td>
<td></td>
</tr>
<tr>
<td>mill</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. STEEL</td>
<td>FE</td>
<td>5.00000E-02</td>
<td>CR</td>
<td>1.00000E-02</td>
</tr>
<tr>
<td>2. FUEL</td>
<td>NU-23</td>
<td>8.10000E-03</td>
<td>U-238</td>
<td>1.25000E-02</td>
</tr>
<tr>
<td>3. SCRAM</td>
<td>NA-23</td>
<td>2.50000E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>assign</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. CORE</td>
<td>FUEL</td>
<td>3.50000E-03</td>
<td>SCRAM</td>
<td>4.00000E-01</td>
</tr>
<tr>
<td>2. REPLIC</td>
<td>SCRAM</td>
<td>7.00000E-03</td>
<td>STEEL</td>
<td>3.00000E-01</td>
</tr>
</tbody>
</table>

*key start mix card as *

*key end mix card as *

*key end block v read-main*

*key end block v read-end*

*key end input module*
APPENDIX A: SAMPLE INPUT

---

**Sample Problem: Output Description**

-----------------------------

this test problem run on 03/24/95 with solver version 12-06-94 product release 2.6x machine e

**SAMPLE PROBLEM FOR TWOHEX USER'S GUIDE**

**STANDARD K CALCULATION, ALL INPUT BY MEANS OF ASCII TEXT**

```
...block v -- solver input...

---

**NEW AS**
Input defaulted

---

...required input(array name = solin)...

1 1 lev 0/1 - type of calculation
0 0 ist 0 - homogeneous source
0 0 lth 0/1 - direct/adjoint - mode of calculation (default-direct)
0 4 fil 2/4 - left boundary condition 
vacuum/rotational
0 2 fhr 2 - right boundary condition 
vacuum
0 2 iht 2 - top boundary condition 
vacuum/rotational
0 4 fbb 2/4 - bottom boundary condition 
vacuum/rotational

...convergence controls(array name = iter)...

0.0000e+00 1.0000e-04 epel inner iteration convergence criterion (default=0.0001)
0 20 delta maximum number of outer iterations (default=20)
0 0 lim iteration time limit (seconds)

---

```

---

...block v -- solver input (continued)...

---

**NEW AS**
Input defaulted

---

...miscellaneous parameters(array name = misc)...

9.0000e+01 9.0000e+00 bhtg buckling height
1.0000e+00 1.0000e+00 nrm normalization factor
0 0 infl 0/1 no/yes - read input flux from file rflux (auxflux for adjoint)
0 0 ines 0/1 no/yes - read input sources from file itsrc
0 1 igcd 0/1/2/3 - quadrature point arrangement 
 default/convex/triangle/rectangular

...output controls(array name = solout)...

0 fluxp 0/1/2 none/leptotropic/all moments - flux print
0 excep 0/1/2 none/principal/all - macroscopic cross section print
0 fise 0/1 nocity = print final fission source rate
0 ssrc 0/1/2/3 no/yes read/normalized/orth - print (inhomogeneous source
0 ssrc a 0/1 nocy = print angular fluxes (unimplemented)
0 refl 0/1 nocy = write angular fluxes to file rflux (unimplemented)
0 balp 0/1 nocy = print coarse mesh balances (currently unimplemented)

...parameters inferred from input arrays...

2 inci 0/1/2 none/nine/chisome/chis
0 jexd 0/1/2 none/k density vector/hill matrix
0 jexy 0/1/2 no/yes - use y density vector
0 jexn source anisotropy
0 jexn number of source vectors input
0 jexn number of source vectors output
0 jexm number of source vectors input
0 jexm number of source vectors output
0 jexm number of source vectors input
0 jexm number of source vectors output

```

---

5-68 TWOHEX USER'S GUIDE Version 3.0
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Description

******************************************************************************
|
| 9 igem 9 triangular xy
| 2 ngap number of energy groups
| 4 lam angular quadrature order
| 3 xe number of permanent materials
| 2 name number of zones
| 12 it number of triangles per row
| 6 jc number of rows of triangles

******************************************************************************

29 words lcm required 2.1.1

...material assignments to zones...

"key start mats to zones "

zone cross section = sum over mats in the zone of (matl cross section)*(c0 + c1*mod)

which

c0 = 0.0000000000

c1 = 0.0000000000

<table>
<thead>
<tr>
<th>entry</th>
<th>zone</th>
<th>material</th>
<th>c0</th>
<th>c1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CORE</td>
<td>FUEL</td>
<td>3.5000000E-01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>CORE</td>
<td>SOLUTION</td>
<td>2.0000000E-01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>3</td>
<td>CORE</td>
<td>STEEL</td>
<td>1.5000000E-01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>4</td>
<td>REPL</td>
<td>SOLUTION</td>
<td>7.5000000E-01</td>
<td>0.0000000E+00</td>
</tr>
<tr>
<td>5</td>
<td>REPL</td>
<td>STEEL</td>
<td>3.0000000E-01</td>
<td>0.0000000E+00</td>
</tr>
</tbody>
</table>

lcm storage summary...

| total lcm required for this problem | 1623 |
| maximum lcm available | 40000 |

lcm storage summary...

| total lcm required for this problem | 1418 |
| maximum lcm specified | 14000 |

1418 words lcm required 2.1.2

zero flux

***** 4 this is the constant set igamil

mu       eta       xi       phi       weight

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8143935E-02</td>
<td>0.4702161E-02</td>
<td>0.3399810E-02</td>
<td>30.0</td>
</tr>
<tr>
<td>2</td>
<td>0.4402694E-03</td>
<td>0.2541870E-03</td>
<td>0.3399862E-03</td>
<td>30.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0000000E+00</td>
<td>0.9404322E-00</td>
<td>0.3399810E-02</td>
<td>90.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0000000E+00</td>
<td>0.5083740E+00</td>
<td>0.6811363E-03</td>
<td>90.0</td>
</tr>
<tr>
<td>5</td>
<td>-0.8143935E-02</td>
<td>0.4702161E-02</td>
<td>0.3399810E-02</td>
<td>150.0</td>
</tr>
<tr>
<td>6</td>
<td>-0.4402694E-03</td>
<td>0.2541870E-03</td>
<td>0.3399862E-03</td>
<td>150.0</td>
</tr>
</tbody>
</table>

******************************************************************************
|
|... cross section related data from file names 03/24/17 08: version 1 ...

******************************************************************************
|
| 1 STEEL 2 FUEL 3 SOLUTION
## Sample Problem: Output Description

### Table: Neutron Group Constants and Flux Values

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Neutron Flux (cm⁻²·s⁻¹)</th>
<th>k-eff</th>
<th>Max Power</th>
<th>Dominance Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.11331547</td>
<td>3.5</td>
<td>1.11331547</td>
<td>5.8280-06</td>
</tr>
<tr>
<td>2</td>
<td>1.11331547</td>
<td>3.5</td>
<td>1.11331547</td>
<td>5.8280-06</td>
</tr>
<tr>
<td>3</td>
<td>1.11331547</td>
<td>3.5</td>
<td>1.11331547</td>
<td>5.8280-06</td>
</tr>
</tbody>
</table>

---

### Additional Information

- **k-eff** represents the effective multiplication factor for the system.
- **Max Power** indicates the maximum power level in each iteration.
- **Dominance Ratio** shows the relative importance of each group in the total reactor power.
edit run on 03/24/95 with solver version 01-25-95*product release 2.6ex machine e

...edit output...

...block vi - edit specification data...

*key start edit output *

...cross section balancing (balas.re.0)
or
*transport correction (trans-diag, cesaco, or ths)
will NOT be reflected in edits

...input control integers...

1 ptup 0/1 no/yes - print edits desired
0 aned 0/1 no/yes - zone edits desired
0 ajed 0/1 direct/adjoint edit (use rtflu/rtflu file)
0 injmpd 0/1/2/3 print totals only/print broad groups only/same as 1/print all groups and totals
0 byclp 0/1 no/yes - multiply point reaction rates by mesh volumes
0 rtflu 0/1 no/yes - write the rtflu file (zone average flux file)
0 rmflx 0/1 no/yes - write the rmflx file (zone average flux moments file)

...floating parameters...

0.00000000-00 power 0/p no/normalize all results, including flux files, to p power
2.10000000-02 mpsmr rev per fission (default: 210 rev)

...energy related edit information...

2 number of fine neutron groups
0 number of fine groups
1 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/adj/dum/both zones main edit
(requires atomic weights to be present)

...edit output...

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

...description of cross section edits...

<table>
<thead>
<tr>
<th>isotope no</th>
<th>material no</th>
<th>constituent no</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>read</td>
<td>none</td>
</tr>
</tbody>
</table>

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

position

type no
<table>
<thead>
<tr>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>chi</td>
<td>2</td>
</tr>
<tr>
<td>numgrf</td>
<td>3</td>
</tr>
<tr>
<td>total</td>
<td>4</td>
</tr>
<tr>
<td>abs</td>
<td>5</td>
</tr>
</tbody>
</table>
APPENDIX A: SAMPLE INPUT

<table>
<thead>
<tr>
<th>Sample Problem: Output Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

---

**Key start materials**

---

**reaction rates from materials**

---

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

---

*point chi edit for x-plane 1, and broad group 3, of macroscopic cross sections.*

---

*all entries of this array equal 0.00000000*

---

*point runsig edit for x-plane 1, and broad group 3, of macroscopic cross sections.*

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---

**Component number 1**

---
## APPENDIX A: SAMPLE INPUT

<table>
<thead>
<tr>
<th>Sample Problem: Output Description</th>
</tr>
</thead>
</table>

| 3 | 3.684510E-04 | 2.86598E-04 | 2.89667E-04 | 1.80713E-04 | 1.50488E-04 | 0.00000E+00 | 0.00000E+00 | 0.00000E+00 |
| 2 | 4.894711E-04 | 4.27508E-04 | 4.18423E-04 | 3.17492E-04 | 2.80667E-04 | 1.80713E-04 | 0.00000E+00 | 0.00000E+00 |
| 1 | 5.55767E-04 | 5.12160E-04 | 4.89471E-04 | 4.27508E-04 | 3.68451E-04 | 2.86598E-04 | 2.12160E-04 | 0.00000E+00 |

<table>
<thead>
<tr>
<th>x-mash 9 x-mash 10 x-mash 11 x-mash 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

---

### run highlights

- all modules are tentatively go.
- interface file generated.
- cross sections from cards.
- interface module files written.
- interface file again written.
- interface file assign written.
- interface file results written.
- interface file edit written.
- edit execution.
- edit completed.

### storage and timing history

<table>
<thead>
<tr>
<th>module</th>
<th>words</th>
<th>limit</th>
<th>words</th>
<th>limit</th>
<th>seconds</th>
<th>seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.8</td>
<td>0.4</td>
</tr>
<tr>
<td>100</td>
<td>3662</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.1</td>
<td>0.2</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.6</td>
<td>0.0</td>
</tr>
<tr>
<td>102</td>
<td>192</td>
<td>97</td>
<td>40000</td>
<td>79</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>103</td>
<td>50</td>
<td>40000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>2066</td>
<td>0</td>
<td>40000</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>191</td>
<td>40000</td>
<td>83</td>
<td>140000</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>108</td>
<td>547</td>
<td>0</td>
<td>40000</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>3677</td>
<td>0</td>
<td>40000</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>112</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>1633</td>
<td>1418</td>
<td>40000</td>
<td>140000</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>23</td>
<td>259</td>
<td>0</td>
<td>40000</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>29</td>
<td>661</td>
<td>0</td>
<td>40000</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>301</td>
<td>1666</td>
<td>721</td>
<td>40000</td>
<td>140000</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>302</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

...execution terminated...

---
On UNIX or UNICOS systems, the input is on STDIN and the printed output is on STDERR. 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. \n
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

```bash
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>
<thead>
<tr>
<th>LIB</th>
<th>SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>Current Working Directory (CWD).</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>SNXSPATH, then CWD, but see text below.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>None, the library is contained in the input file.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>CWD</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>CWD</td>
</tr>
<tr>
<td>MENDF*a</td>
<td>Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.</td>
</tr>
<tr>
<td>MENDFg*b</td>
<td>Path defined in the code on UNICOS.</td>
</tr>
<tr>
<td>XSLIB</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>other</td>
<td>For any name other than those above, the program will assume the form is XSLIB and search for it in SNXS-PATH, then CWD.</td>
</tr>
</tbody>
</table>

*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.
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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>6-5</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>6-7</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>6-9</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>6-11</td>
</tr>
<tr>
<td>DOCUMENTATION FOR THREEDANT USAGE</td>
<td>6-13</td>
</tr>
<tr>
<td>What Is In This User’s Guide</td>
<td>6-13</td>
</tr>
<tr>
<td>What Is Available Elsewhere</td>
<td>6-14</td>
</tr>
<tr>
<td>THREEDANT INPUT OVERVIEW</td>
<td>6-17</td>
</tr>
<tr>
<td>Input Block Order</td>
<td>6-17</td>
</tr>
<tr>
<td>Free Field Input Summary</td>
<td>6-19</td>
</tr>
<tr>
<td>Arrays</td>
<td>6-19</td>
</tr>
<tr>
<td>Numeric Data Items</td>
<td>6-19</td>
</tr>
<tr>
<td>Character Data Items</td>
<td>6-19</td>
</tr>
<tr>
<td>Blocks</td>
<td>6-20</td>
</tr>
<tr>
<td>Strings</td>
<td>6-20</td>
</tr>
<tr>
<td>Comments</td>
<td>6-20</td>
</tr>
<tr>
<td>Operators</td>
<td>6-20</td>
</tr>
<tr>
<td>Frequently Used Operators</td>
<td>6-21</td>
</tr>
<tr>
<td>MINI-MANUAL Introduction</td>
<td>6-22</td>
</tr>
<tr>
<td>MINI MANUAL</td>
<td>6-23</td>
</tr>
<tr>
<td>THREEDANT INPUT DETAILS</td>
<td>6-29</td>
</tr>
<tr>
<td>Introduction</td>
<td>6-29</td>
</tr>
<tr>
<td>Title Line Details</td>
<td>6-32</td>
</tr>
<tr>
<td>Title Line Control</td>
<td>6-32</td>
</tr>
<tr>
<td>Block-I Details: Dimensions and Controls</td>
<td>6-33</td>
</tr>
<tr>
<td>Dimensions</td>
<td>6-33</td>
</tr>
<tr>
<td>Storage Requirements</td>
<td>6-34</td>
</tr>
<tr>
<td>Run Configuration Controls</td>
<td>6-35</td>
</tr>
<tr>
<td>Block-II Details: Geometry</td>
<td>6-36</td>
</tr>
<tr>
<td>Geometry Arrays</td>
<td>6-36</td>
</tr>
<tr>
<td>Block-III Details: Nuclear Data</td>
<td>6-37</td>
</tr>
<tr>
<td>Nuclear Data Type and Options</td>
<td>6-37</td>
</tr>
<tr>
<td>Alternate Library Name</td>
<td>6-39</td>
</tr>
<tr>
<td>Text Cross-Section Library Format</td>
<td>6-41</td>
</tr>
<tr>
<td>Block-IV Details: Cross-Section Mixing</td>
<td>6-43</td>
</tr>
<tr>
<td>MATLS input array</td>
<td>6-44</td>
</tr>
<tr>
<td>Primary Mixing Arrays</td>
<td>6-44</td>
</tr>
<tr>
<td>ASSIGN input array</td>
<td>6-45</td>
</tr>
<tr>
<td>PREMIX input array</td>
<td>6-45</td>
</tr>
<tr>
<td>Character Names vs. Numeric Names</td>
<td>6-46</td>
</tr>
<tr>
<td>Mixing Array for a Concentration Search</td>
<td>6-46</td>
</tr>
<tr>
<td>ASGMOD input array</td>
<td>6-47</td>
</tr>
<tr>
<td>Concentration Modifier</td>
<td>6-47</td>
</tr>
<tr>
<td>Fine Mesh Mixing</td>
<td>6-48</td>
</tr>
<tr>
<td>Miscellaneous Mixing Input</td>
<td>6-49</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

**Block-V Details: Solver Input** ................................................................. 6-50  
- Desired Calculation ................................................................. 6-50  
- Iteration Controls ................................................................. 6-51  
- K-Code Convergence ........................................................... 6-51  
- Output Controls ................................................................. 6-52  
- Miscellaneous Solver Input ................................................... 6-53  
- Quadrature Details ............................................................... 6-54  
- Flux Guess From a File............................................................... 6-55  
- General Eigenvalue Search Control ......................................... 6-55  
- Dimension Search Input ......................................................... 6-56  
- Concentration Search Input ..................................................... 6-56  
- Volumetric Source Options ..................................................... 6-57  
- Boundary Source Input ............................................................ 6-58  
- Boundary Source Vector Input Combinations ......................... 6-60  

**Block-VI Details: Edit Input** ................................................................. 6-61  
- Edit Spatial Specifications ....................................................... 6-61  
- Reaction Rates from Cross Sections ......................................... 6-62  
- Edit Cross-Section Types by Position and Name ....................... 6-64  
- Reaction Rates from User Response Functions ......................... 6-65  
- Energy Group Collapse Specifications .................................. 6-66  
- Reaction Rate Summing ............................................................ 6-67  
- Mass Inventories ................................................................. 6-67  
- Power Normalization ............................................................. 6-68  
- Miscellaneous Edit Items ......................................................... 6-69  
- Special Plot Linkage .............................................................. 6-70  
- MENDF Library Edit Cross Sections ....................................... 6-71

**REFERENCES** ...................................................................................... 6-73

**APPENDIX A: SAMPLE INPUT** ............................................................ 6-75  
- Sample Problem: Standard \( \kappa_{\text{eff}} \) Calculation .................. 6-75  
- Sample Problem: Output Description ....................................... 6-75  
- Sample Problem: Output Listing ............................................. 6-77

**APPENDIX B: OPERATING SYSTEM SPECIFICS** .................................. 6-97  
- UNIX/UNICOS Execution ....................................................... 6-97  
- Library Search Path .............................................................. 6-98
LIST OF FIGURES

Figure 6.1: THREEDANT Input Order ................................................................. 6-18
Figure 6.2: Orientation of Faces ................................................................. 6-59
LIST OF TABLES

Table 6.1: LIBNAME Availability ................................................................. 6-39
Table 6.2: UNIX Search Path ............................................................... 6-98
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.\textsuperscript{1,2} It is a three-dimensional version of the one-dimensional code, ONEDANT.\textsuperscript{3}

THREEDANT\textsuperscript{TM} is based on the modular construction of the DANTSYS\textsuperscript{TM} 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\textsuperscript{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\textsuperscript{5} to accelerate the iterative process in the SOLVER module,
4. direct (forward) or adjoint calculational capability,
5. x-y-z and r-z-\theta geometry options,
6. arbitrary anisotropic scattering order,
7. vacuum, reflective, periodic, white, or surface source boundary condition options,
8. inhomogeneous (fixed) source or $k_{\text{eff}}$ calculation options, as well as time-absorption ($\alpha$), 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.
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.
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:

\[
\text{SOURCX} = 0 \ 0 \ 0 \ 1.1 \ 1.1 \ 0 \ 0 \ 0 \ 0
\]

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:

\[ \text{SOURCX}=0\ 0\ 0\ 2\ R\ 1.1\ F0 \]

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 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

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

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*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
(316 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
ITIL
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]

--- 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: ----

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: ----

Options 3b: ----
BSRITG [NGROUP] BSRITYZ [JT;KT] BSRITA [MM*4]

--- Boundary Source (cont’d) ---

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]  
SABOIT [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≤JT*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
ICRPED

MICSUM [-]
IRSUMS [-]

MASSED

POWER
MEVPER

RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE

PRPLTED
IPLANE
JPLANE
KPLANE

T
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 316)\textsuperscript{a} {Required}

<table>
<thead>
<tr>
<th>Word</th>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NHEAD</td>
<td>Number of title lines that follow.\textsuperscript{b}</td>
</tr>
<tr>
<td>2</td>
<td>NOTTY</td>
<td>Suppress output to on-line user terminal? 0/1 = no/yes.</td>
</tr>
<tr>
<td>3</td>
<td>NOLIST</td>
<td>Suppress listing of all ASCII text input? 0/1 = no/yes. (default=no)</td>
</tr>
</tbody>
</table>

\textsuperscript{a} WARNING! Note that this first line is in fixed format.

\textsuperscript{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

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGEOM</td>
<td>Geometry. (14/15 = x\cdot y\cdot z/r\cdot z\cdot \theta) or use one of the following character strings: (X-Y-Z, R-Z-T)</td>
</tr>
<tr>
<td>NGROUP</td>
<td>Number of energy groups.</td>
</tr>
<tr>
<td>ISN</td>
<td>(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.)</td>
</tr>
<tr>
<td>NISO</td>
<td>Number of physical isotopes on the basic input cross-section library.</td>
</tr>
<tr>
<td>MT</td>
<td>Number of physical materials(^a) to be created.</td>
</tr>
<tr>
<td>NZONE</td>
<td>Number of geometric zones(^b) in problem.</td>
</tr>
<tr>
<td>IM</td>
<td>Number of coarse mesh intervals(^c) in the x (or r) direction.</td>
</tr>
<tr>
<td>IT</td>
<td>Total number of fine mesh intervals(^d) in the x (or r) direction.</td>
</tr>
<tr>
<td>JM</td>
<td>Number of coarse mesh intervals in the y (or z) direction.</td>
</tr>
<tr>
<td>JT</td>
<td>Total number of fine mesh intervals in the y (or z) direction.</td>
</tr>
<tr>
<td>KM</td>
<td>Number of coarse mesh intervals in the z (or (\theta)) direction.</td>
</tr>
<tr>
<td>KT</td>
<td>Total number of fine mesh intervals in the z (or (\theta)) direction.</td>
</tr>
</tbody>
</table>

\(^a\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSCM</td>
<td>Length of SCM desired (default=400000)</td>
</tr>
<tr>
<td>MAXLCM</td>
<td>Length of LCM desired (default=1400000)</td>
</tr>
</tbody>
</table>

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}

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

Note: Default on all these controls is no.
Block-II Details: Geometry

Geometry Arrays\(^a\)
{Required}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMESH [IM+1]</td>
<td>x (or r) coordinates of coarse mesh edges.</td>
</tr>
<tr>
<td>YMESH [JM+1]</td>
<td>y (or z) coordinates of coarse mesh edges.</td>
</tr>
<tr>
<td>ZMESH [KM+1]</td>
<td>z (or θ) coordinates of coarse mesh edges. When r-z-θ geometry used, θ is in revolutions.</td>
</tr>
<tr>
<td>XINTS [IM]</td>
<td>Number of fine meshes in each coarse x (r) mesh</td>
</tr>
<tr>
<td>YINTS [JM]</td>
<td>Number of fine meshes in each coarse y (z) mesh</td>
</tr>
<tr>
<td>ZINTS [KM]</td>
<td>Number of fine meshes in each coarse z (θ) mesh</td>
</tr>
<tr>
<td>ZONES [IM;JM*KM]</td>
<td>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.</td>
</tr>
</tbody>
</table>

---

\(^a\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LIB</strong></td>
<td>Name and form of the cross-section data file. Enter as a data item one of the following words:</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>ISOTXS$^b$</td>
<td>CCC standard isotope ordered binary cross-section file.</td>
</tr>
<tr>
<td>XSLIB</td>
<td>ASCII text library supplied in a separate file named XSLIB.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>ASCII text library follows after this block of input (after the T of Block-III).</td>
</tr>
<tr>
<td>GRUPXS$^c$</td>
<td>CCC standard group ordered cross-section file.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Binary library supplied as a separate file named BXSLIB. [See “Binary Form of Card-Image Libraries (the BXSLIB file)” on page 10-12. 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.</td>
</tr>
<tr>
<td>MACRXS$^d$</td>
<td>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.</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>See “XSLIBB Card-Image Library File” on page 10-12.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>ASCII form of MACRXS file.</td>
</tr>
<tr>
<td>MENDF</td>
<td>(LANL only) See “The Los Alamos MENDF5 Cross-Section Library” on page 10-13.</td>
</tr>
<tr>
<td>MENDFG</td>
<td>(LANL only) See “The Los Alamos MENDF5G Gamma Cross-Section Library” on page 10-14.</td>
</tr>
<tr>
<td><em>other</em></td>
<td>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.</td>
</tr>
</tbody>
</table>
### Nuclear Data Type and Options (Cont.)

#### {Required}

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

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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBNAME</td>
<td>Alternate name of the library file. May be used only with certain types of libraries. See Table 6.1.</td>
</tr>
</tbody>
</table>

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 6.1.

Table 6.1 LIBNAME Availability

<table>
<thead>
<tr>
<th>LIB</th>
<th>LIBNAME AVAILABLE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>No</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>Yes</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>Yes</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>ODNINP</td>
<td>No</td>
</tr>
<tr>
<td>MACBCD</td>
<td>No</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>No</td>
</tr>
<tr>
<td>MENDFa</td>
<td>No</td>
</tr>
<tr>
<td>MENDFGb</td>
<td>No</td>
</tr>
<tr>
<td>XSLIB</td>
<td>Yes</td>
</tr>
<tr>
<td>other</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

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}

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

\[ \cdots \sigma_{\text{abs}}, \nu\sigma_{T}, \sigma_{\text{total}}, \cdots \sigma_{g+2 \rightarrow g}, \sigma_{g+1 \rightarrow g}, \sigma_{g \rightarrow g}, \sigma_{g-1 \rightarrow g}, \sigma_{g-2 \rightarrow g}, \cdots \]

Notice that all terms in the scattering matrix are in positions relative to that of the selfscattering 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 \( \Sigma_g \) 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 \( \Sigma_g \) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATLS(a) ([-;MT])</td>
<td>Instructions for mixing “isotopes” or premixes into “materials.” See details below.</td>
</tr>
<tr>
<td>ASSIGN(b) ([-;NZONE])</td>
<td>Assignments of materials to geometric zones. See below.</td>
</tr>
<tr>
<td>PREMIX ([-;\text{optional}])</td>
<td>Instructions for mixing “isotopes” into premixes. See below.</td>
</tr>
</tbody>
</table>

\(a\). 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:

\[
\text{MATLS} = \text{mat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \text{ etc.};
\]

where \(\text{mat}_1\) is the desired character name of the first material and \(\text{comp}_1, \text{comp}_2, \text{and so on}\) are the character names of its components which have “densities” of, respectively, \(\text{den}_1, \text{den}_2\), and so on. Additional materials (i.e., \(\text{mat}_2, \text{mat}_3, \text{and so on up to the required number, } \text{MT}\)) 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 \( \text{den}_i \) 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:

\[ \text{MATLS=} \quad \text{mat}_1 \text{ comp}_1; \quad \text{mat}_2 \text{ comp}_2 \text{ den}_2; \quad \ldots \text{etc.} \ldots ; \]

**ASSIGN Input array**

The general form of the ASSIGN instruction is shown below:

\[ \text{ASSIGN=} \quad \text{zone}_1 \text{ mat}_1 \text{ vol}_1, \text{ mat}_2 \text{ vol}_2, \ldots \text{etc.} \ldots ; \]

where \( \text{zone}_i \) is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). \( \text{mat}_1, \text{ mat}_2, \) and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" \( \text{vol}_1, \text{ vol}_2, \) and so on. Additional zones (i.e., \( \text{zone}_2, \text{ zone}_3, \) 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:

\[ \text{PREMIX=} \quad \text{mat}_1 \text{ comp}_1 \text{ den}_1, \text{ comp}_2 \text{ den}_2, \ldots \text{etc.} \ldots ; \]
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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASGMODa</td>
<td>$C_1$ parameters used in concentration searches. See the discussion below.</td>
</tr>
</tbody>
</table>

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:

\[ C(z,x) = C_0(z,x) + C_1(z,x) \times \text{CMOD} \]

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:

\[ \text{ASGMOD} = \text{zone mat}_m \text{vol}_m, \text{mat}_n \text{vol}_n, \ldots \text{etc.} \ldots ; \]

where **zone** is the character name of any zone in the problem, **mat** and **vol** indicate the character names of any of the materials that will be present in this zone, and **vol** and **vol** indicate the \( C_1 \) values for respectively, **mat** and **vol**, 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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOD</td>
<td>Concentration modifier. Input value is not used in a search. See the discussion below.</td>
</tr>
</tbody>
</table>

---

THREEDANT USER'S GUIDE Version 3.0
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
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FMMIX</td>
<td>Read the composition of each fine mesh from the file LNK3DNT. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATNAM [MT]</td>
<td>Character material names for Materials. Used only if the <code>mat</code> 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.</td>
</tr>
<tr>
<td>ZONNAM [NZONE]</td>
<td>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.</td>
</tr>
<tr>
<td>MATSPEC [≤MT]</td>
<td>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: <code>ATDENS</code> (default) atomic densities <code>ATFRAC</code> atomic fractions <code>WTFRAC</code> 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.</td>
</tr>
<tr>
<td>ATWT [≤2*NISO] (required(^{b}))</td>
<td>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: [\text{ATWT} = is_{o1} atwt_{1} is_{o2} atwt_{2} \ldots] where (is_{o_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].</td>
</tr>
</tbody>
</table>

---

\(^{a}\) 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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IEVT</strong></td>
<td>Calculation type: Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>0</td>
<td>source</td>
</tr>
<tr>
<td>1</td>
<td>$k_{\text{eff}}$</td>
</tr>
<tr>
<td>2</td>
<td>$\alpha$ (time absorption) search</td>
</tr>
<tr>
<td>3</td>
<td>concentration search</td>
</tr>
<tr>
<td>4</td>
<td>dimension search</td>
</tr>
<tr>
<td><strong>ISCT</strong></td>
<td>Legendre order of scattering (default = 0).</td>
</tr>
<tr>
<td><strong>ITH</strong></td>
<td>$0/1$ = direct/adjoint calculation (default = 0).</td>
</tr>
<tr>
<td><strong>IBL</strong></td>
<td>Left boundary condition(^a): 0/1/3 = vacuum/reflective/white (default = vacuum).</td>
</tr>
<tr>
<td><strong>IBR</strong></td>
<td>Right boundary condition: 0/1/3 = vacuum/reflective/white (default = vacuum).</td>
</tr>
<tr>
<td><strong>IBT</strong></td>
<td>Top boundary condition: 0/1/2/3 = vacuum/reflective/periodic/white. (default = vacuum).</td>
</tr>
<tr>
<td><strong>IBB</strong></td>
<td>Bottom boundary condition: 0/1/2/3 = vacuum/reflective/periodic/white. (default = vacuum).</td>
</tr>
<tr>
<td><strong>IBFRNT</strong></td>
<td>Front boundary condition: 0/1/2/3 = vacuum/reflective/periodic/white. (default = vacuum).</td>
</tr>
<tr>
<td><strong>IBBACK</strong></td>
<td>Back boundary condition: 0/1/2/3 = vacuum/reflective/periodic/white. (default = vacuum).</td>
</tr>
</tbody>
</table>

\(^a\) The left boundary condition applies only for x-y-z geometry.
Iteration Controls
{Required}

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

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 \((\frac{1}{k_{eff}})^{1/N_{A}}\Phi\) where \(\Phi\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>KCALC</td>
<td>Special Criticality Convergence Scheme. 0/1 = no/yes.</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLUXP</td>
<td>Final flux print. 0/1/2 = no/isotropic/all moments.</td>
</tr>
<tr>
<td>XSECTP</td>
<td>Cross-section print. 0/1/2 = no/principal/all.</td>
</tr>
<tr>
<td>FISSRP</td>
<td>Fission source rate print. 0/1 = no/yes.</td>
</tr>
<tr>
<td>SOURCP</td>
<td>Source print. 0/1/2/3 = no/as input/normalized/both.</td>
</tr>
<tr>
<td>ANGP</td>
<td>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.</td>
</tr>
<tr>
<td>BALP</td>
<td>Coarse Mesh Interval Print Options. Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>RAFLUX</td>
<td>Prepare angular flux file (RAFLXM or AAFLXM). 0/1 = no/yes.</td>
</tr>
<tr>
<td>RMFLUX</td>
<td>Prepare flux moments file (RMFLUX or AMFLUX). 0/1 = no/yes.</td>
</tr>
</tbody>
</table>
# Miscellaneous Solver Input

## {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRCOR</td>
<td>Apply transport correction(^a) to cross sections on MACRXS file. Enter one of the following words:</td>
</tr>
<tr>
<td></td>
<td><strong>Word</strong></td>
</tr>
<tr>
<td></td>
<td>DIAG</td>
</tr>
<tr>
<td></td>
<td>BHS</td>
</tr>
<tr>
<td></td>
<td>CESARO</td>
</tr>
<tr>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>NORM</td>
<td>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(<em>{\text{eff}}) problems where the integral is equal to NORM(k</em>{\text{eff}}).) Any fluxes printed here (i.e., caused by setting FLUXP nonzero) will be normalized consistently with this source rate.</td>
</tr>
<tr>
<td>CHI [NGROUP;M]</td>
<td>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.</td>
</tr>
<tr>
<td>DEN [IT;JT*KT]</td>
<td>Density factor to use at each fine mesh point.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>DENX [IT](^c)</td>
<td>Density factor to use at each fine x-mesh (default=1).</td>
</tr>
<tr>
<td>and/or</td>
<td></td>
</tr>
<tr>
<td>DENY [JT]</td>
<td>Density factor to use at each fine y-mesh (default=1).</td>
</tr>
<tr>
<td>and/or</td>
<td></td>
</tr>
<tr>
<td>DENZ [KT]</td>
<td>Density factor to use at each fine z-mesh (default=1).</td>
</tr>
<tr>
<td>WDAMP [NGROUP]</td>
<td>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.</td>
</tr>
</tbody>
</table>

---

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:

\[
\text{DEN}(i,j,k) = \text{DENX}(i) \times \text{DENY}(j) \times \text{DENZ}(k)
\]

d. Recommend 1.0 < W < 4.0 for shielding applications.
# Quadrature Details

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRPSN [NGROUP]^a</td>
<td>$S_n$ order to be used for each group.</td>
</tr>
<tr>
<td>IQUAD</td>
<td>Source of the quadrature constants.</td>
</tr>
<tr>
<td></td>
<td>Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td><strong>Value</strong></td>
</tr>
<tr>
<td></td>
<td>-3</td>
</tr>
<tr>
<td></td>
<td>-2</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>WGT^b [MM^c]</td>
<td>Quadrature weights.</td>
</tr>
<tr>
<td></td>
<td>[optional]</td>
</tr>
<tr>
<td>MU [MM]</td>
<td>Mu cosines.</td>
</tr>
<tr>
<td></td>
<td>[optional]</td>
</tr>
<tr>
<td>ETA [MM]</td>
<td>Eta cosines.</td>
</tr>
<tr>
<td></td>
<td>[optional]</td>
</tr>
</tbody>
</table>

---

^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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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
{ IEVT >1 }

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

\(^a\) 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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XM [IM]</td>
<td>x-dimension fractional change per coarse mesh.</td>
</tr>
<tr>
<td>YM [JM]</td>
<td>y-dimension fractional change per coarse mesh.</td>
</tr>
<tr>
<td>ZM [KM]</td>
<td>z-dimension fractional change per coarse mesh.</td>
</tr>
</tbody>
</table>

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:

\[
XMESH_{i+1} = XMESH_i + \{XMESH_{i+1}-XMESH_i\} \times \left[1.0 + EV \times XM_i\right], \quad i = 1, \ldots, IM
\]

\[
YMESH_{j+1} = YMESH_j + \{YMESH_{j+1}-YMESH_j\} \times \left[1.0 + EV \times YM_j\right], \quad j = 1, \ldots, JM
\]

\[
ZMESH_{k+1} = ZMESH_k + \{ZMESH_{k+1}-ZMESH_k\} \times \left[1.0 + EV \times ZM_k\right], \quad k = 1, \ldots, KM
\]

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}**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\) 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)

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSORS</td>
<td>Read source from interface file FIXSRC. 0/1 = no/yes</td>
</tr>
</tbody>
</table>

----- 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:**

- **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 \(\theta\)) spatial distribution for each moment.

(Spectrum is assumed to be flat with value unity)

**Option 3:**

- **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 \(\theta\)) spatial distribution for each moment.

**Option 4:**

- **SOURCF [IT;JT*KT*NGROUP*NMQ]**: Spatial distribution for each row, group, and moment.

**Option 5:**

- **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:

$$\text{SOURCF}(i,j,k,g,m) = \text{SOURCE}(g,m) \times \text{SOURCX}(i,m) \times \text{SOURCY}(j,m) \times \text{SOURCZ}(k,m)$$

Boundary Source Input
{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>----</td>
<td>For a text-input source, choose one of the following options:</td>
</tr>
</tbody>
</table>

Option 1: Isotropic Boundary Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SILEFT</td>
<td>Isotropic source on the left face. (Spectrum at each y,z mesh interval.)</td>
</tr>
<tr>
<td>SIRITE</td>
<td>Isotropic source on the right face.</td>
</tr>
<tr>
<td>SIBOTT</td>
<td>Isotropic source on the bottom face.</td>
</tr>
<tr>
<td>SITOP</td>
<td>Isotropic source on the top face.</td>
</tr>
<tr>
<td>SIFRNT</td>
<td>Isotropic source on the front face.</td>
</tr>
<tr>
<td>SIBACK</td>
<td>Isotropic source on the back face.</td>
</tr>
</tbody>
</table>

Option 2: Full Angular Boundary Source

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAILFET</td>
<td>Angular flux on the left for each angle, group, and y,z mesh interval.</td>
</tr>
<tr>
<td>SARIETE</td>
<td>Angular fluxes on the right face.</td>
</tr>
<tr>
<td>SABOTT</td>
<td>Angular fluxes on the bottom face.</td>
</tr>
<tr>
<td>SATOP</td>
<td>Angular fluxes on the top face.</td>
</tr>
<tr>
<td>SAFRNT</td>
<td>Angular fluxes on the front face</td>
</tr>
<tr>
<td>SABACK</td>
<td>Angular fluxes on the back face</td>
</tr>
</tbody>
</table>

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: $\mu < 0$, $\eta < 0$, $\tau < 0$; $\mu > 0$, $\eta < 0$, $\tau < 0$; $\mu < 0$, $\eta > 0$, $\tau < 0$; $\mu > 0$, $\eta < 0$, $\tau > 0$; $\mu < 0$, $\eta > 0$, $\tau > 0$; $\mu > 0$, $\eta > 0$, and $\mu > 0$, $\eta > 0$, $\tau > 0$, where each angular boundary source requires four octants for specification.

b. See "Quadrature Details" on page 6-54 for value of MM.
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- \[ S(m,g,y,z) = BSLFTG(g) * SALEFT(m,y,z) \]

Figure 6.2 Orientation of Faces
## Boundary Source Vector Input Combinations

### {Optional}

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Bottom</th>
<th>Top</th>
<th>Front</th>
<th>Back</th>
</tr>
</thead>
</table>

### Option 3b: Boundary Source By Product Of A Spectrum Vector, An Angular Distribution Vector, And A 2d Spatial Array

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Bottom</th>
<th>Top</th>
<th>Front</th>
<th>Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSLFTY [JT;KT]</td>
<td>BSRITY [JT;KT]</td>
<td>BSBOTXZ [IT;KT]</td>
<td>BSTOPXZ [IT;KT]</td>
<td>BSFRNXY [IT;JT]</td>
<td>BSBACYX [IT;JT]</td>
</tr>
</tbody>
</table>

### Option 3c: Boundary Source By Product Of A Spectrum Vector And A 3d Space-Angle Array

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Bottom</th>
<th>Top</th>
<th>Front</th>
<th>Back</th>
</tr>
</thead>
</table>

### Option 3d: Boundary Source By Product Of A Spectrum Vector And A 3d Angle-Space Array

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>Bottom</th>
<th>Top</th>
<th>Front</th>
<th>Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>SALEFT [MM<em>4; JT</em>KT]</td>
<td>SARITT [MM<em>4; JT</em>KT]</td>
<td>SABOTT [MM<em>4; IT</em>KT]</td>
<td>SATOPT [MM<em>4; IT</em>KT]</td>
<td>SAFRNT [MM<em>4; IT</em>JT]</td>
<td>SABAKT [MM<em>4; IT</em>JT]</td>
</tr>
</tbody>
</table>
Block-VI Details: Edit Input*

Edit Spatial Specifications
{Required³}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTED</td>
<td>Do edits by fine mesh. 0/1 = no/yes.</td>
</tr>
<tr>
<td>ZNED</td>
<td>Do edits by zone. 0/1 = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below.)</td>
</tr>
<tr>
<td>POINTS [≤IT<em>JT</em>KT] {optional}</td>
<td>Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points)</td>
</tr>
<tr>
<td>EDZONE [IT;JT*KT] {optional}</td>
<td>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)</td>
</tr>
</tbody>
</table>

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

*Optional* 

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDXS [≤NEDT]</td>
<td>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)</td>
</tr>
<tr>
<td>RESDNT</td>
<td>Do edits using the resident macroscopic cross section at each point. 0/1 = no/yes.</td>
</tr>
<tr>
<td>EDISOS [≤NISO]</td>
<td>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).</td>
</tr>
<tr>
<td>EDCONS [≤NISO]</td>
<td>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).</td>
</tr>
<tr>
<td>EDMATS [≤MT]</td>
<td>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).</td>
</tr>
<tr>
<td>XDF[d] [IT]</td>
<td>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).</td>
</tr>
<tr>
<td>YDF [JT]</td>
<td></td>
</tr>
<tr>
<td>ZDF [KT]</td>
<td></td>
</tr>
</tbody>
</table>

---
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:

\[
XDF(i) \times YDF(j) \times ZDF(k)
\]
## Edit Cross-Section Types by Position and Name

<table>
<thead>
<tr>
<th>CROSS-SECTION INPUT VIA ISOTXS or GRUPXS</th>
<th>CROSS-SECTION INPUT VIA ASCII TEXT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>EDIT</strong></td>
</tr>
<tr>
<td>chi</td>
<td>1</td>
</tr>
<tr>
<td>nu-fission</td>
<td>2</td>
</tr>
<tr>
<td>total</td>
<td>3</td>
</tr>
<tr>
<td>absorption</td>
<td>4</td>
</tr>
<tr>
<td>(n,p)</td>
<td>5</td>
</tr>
<tr>
<td>(n,d)</td>
<td>6</td>
</tr>
<tr>
<td>(n,t)</td>
<td>7</td>
</tr>
<tr>
<td>(n,alpha)</td>
<td>8</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>9</td>
</tr>
<tr>
<td>(n,gamma)</td>
<td>10</td>
</tr>
<tr>
<td>fission</td>
<td>11</td>
</tr>
<tr>
<td>transport</td>
<td>12</td>
</tr>
</tbody>
</table>

<sup>a</sup> Names are eight characters. A period within a name in this table denotes a blank.

<sup>b</sup> 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.

<sup>c</sup> 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\(^a\)}

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

\(^a\) 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:

\[ RSFX(i,m) \times RSFY(j,m) \times RSFZ(k,m) \times RSFE(g,m) \]
## Energy Group Collapse Specifications
### {Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICOLL [NBG]</td>
<td>Edit energy group collapsing option:</td>
</tr>
<tr>
<td></td>
<td>Number of SOLVER energy groups in each EDIT broad group.</td>
</tr>
<tr>
<td></td>
<td>The NBG entries must sum to NGROUP.</td>
</tr>
<tr>
<td></td>
<td>(default = 1 energy group per EDIT broad group)</td>
</tr>
<tr>
<td>IGRPED</td>
<td>Print option on energy groups. Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>0</td>
<td>Print energy group totals only</td>
</tr>
<tr>
<td>1</td>
<td>Print broad groups only</td>
</tr>
<tr>
<td>2</td>
<td>Print broad groups only (same as 1)</td>
</tr>
<tr>
<td>3</td>
<td>Print both broad groups and totals</td>
</tr>
</tbody>
</table>
### Reaction Rate Summing

#### Optional

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MICSUM [≤500 sums]</td>
<td>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).</td>
</tr>
<tr>
<td>IRSUMS [≤500 sums]</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Mass Inventories

#### Optional

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSED</td>
<td>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.</td>
</tr>
</tbody>
</table>
Power Normalization

{Optional}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>Normalize to POWER megawatts.&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>{required}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 6-53).</td>
</tr>
<tr>
<td>MEVPER</td>
<td>MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power.</td>
</tr>
<tr>
<td>{required}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
</tbody>
</table>

<sup>a</sup> Note that this normalization is meaningless if you are using the results of an adjoint run.
### Miscellaneous Edit Items

#### Optional

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>RZFLUX</td>
<td>Write the CCCC standard zone(^a) flux file RZFLUX or AZFLUX.</td>
</tr>
<tr>
<td></td>
<td>(0/1 = \text{no/yes.})</td>
</tr>
<tr>
<td>RZMFLX</td>
<td>Write the code-dependent zone(^b) flux moments file RZMFLX or AZMFLX.</td>
</tr>
<tr>
<td></td>
<td>(0/1 = \text{no/yes.})</td>
</tr>
<tr>
<td>EDOUTF(^c)</td>
<td>ASCII output files control. Enter one of the following values:</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-3</td>
<td>Write both EDTOGX (without scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>-2</td>
<td>Write EDTOGX file (without scalar fluxes).</td>
</tr>
<tr>
<td>0</td>
<td>Write neither file. (default)</td>
</tr>
<tr>
<td>1</td>
<td>Write EDTOUT file.</td>
</tr>
<tr>
<td>2</td>
<td>Write EDTOGX file (with scalar fluxes).</td>
</tr>
<tr>
<td>3</td>
<td>Write both EDTOGX (with scalar fluxes) and EDTOUT files.</td>
</tr>
<tr>
<td>BYVOLP</td>
<td>Printed point reaction rates will have been multiplied by the mesh volume.</td>
</tr>
<tr>
<td></td>
<td>(0/1 = \text{no/yes.})</td>
</tr>
<tr>
<td>AJED(^d)</td>
<td>Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file.</td>
</tr>
<tr>
<td></td>
<td>(0/1 = \text{regular/adjoint.})</td>
</tr>
<tr>
<td>FLUXONE</td>
<td>Flux override. (0/1 = \text{no/yes.})</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

---

\(^a\) 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}

<table>
<thead>
<tr>
<th>Name</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 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

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\chi$</td>
<td>CHI</td>
<td>fission spectrum</td>
</tr>
<tr>
<td>$\nu\sigma_f$</td>
<td>NUSIGF</td>
<td>effective nu-sigma-fission</td>
</tr>
<tr>
<td>$\sigma_t$</td>
<td>TOTAL</td>
<td>Total cross section</td>
</tr>
<tr>
<td>$\sigma_a$</td>
<td>ABS</td>
<td>absorption $^a$</td>
</tr>
<tr>
<td>(n,n)</td>
<td>MEND1</td>
<td>elastic scattering</td>
</tr>
<tr>
<td>(n,n')</td>
<td>MEND2</td>
<td>inelastic scattering</td>
</tr>
<tr>
<td>(n,2n)</td>
<td>MEND3</td>
<td>n,2n scattering</td>
</tr>
<tr>
<td>(n,3n)</td>
<td>MEND4</td>
<td>n,3n scattering</td>
</tr>
<tr>
<td>(n,\gamma)</td>
<td>MEND5</td>
<td>gamma production</td>
</tr>
<tr>
<td>(n,\alpha)</td>
<td>MEND6</td>
<td>alpha production</td>
</tr>
<tr>
<td>(n,p)</td>
<td>MEND7</td>
<td>proton production</td>
</tr>
<tr>
<td>(n,f)</td>
<td>MEND8</td>
<td>direct fission</td>
</tr>
<tr>
<td>(n,n')f</td>
<td>MEND9</td>
<td>second-chance fission</td>
</tr>
<tr>
<td>(n,2n)f</td>
<td>MEND10</td>
<td>third-chance fission</td>
</tr>
<tr>
<td>(n,F)</td>
<td>N-FISS</td>
<td>$[(n,F) = (n,f) + (n,n')f + (n,2n)f]$</td>
</tr>
<tr>
<td>$\chi_p$</td>
<td>MEND12</td>
<td>prompt fission spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(only for fissionable materials)</td>
</tr>
<tr>
<td>$\chi_t$</td>
<td>MEND13</td>
<td>total fission spectrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(only for fissionable materials)</td>
</tr>
</tbody>
</table>

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

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


APPENDIX A: SAMPLE INPUT

Sample Problem: Standard $k_{\text{eff}}$ 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 $k_{\text{eff}}$ 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 $5\times10^{-5}$, 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 $5\times10^{-5}$ 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

Listing of cards in the input stream:

1. 2 0
2. 3D model of a small FHR.
3. Cross sections from input...
4. / Block I...
5. / Block II...
6. / Block III...
7. / Block IV...
8. / Block V...
9. / Block VI...

---

The output listing shows various cards and sections from the input file. The listing includes specific details such as card numbers, model descriptions, and cross sections, indicating a structured approach to the problem.
APPENDIX A: SAMPLE INPUT

Sample Problem: Output Listing

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

...library name...
lib100

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

<table>
<thead>
<tr>
<th>group</th>
<th>chi</th>
<th>vel</th>
<th>lower bound</th>
<th>upper bound</th>
<th>group</th>
<th>chi</th>
<th>vel</th>
<th>lower bound</th>
<th>upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>3</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>4</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
</tbody>
</table>

last neutron group(ing) is number 4

0 halves -1/0/1 - adjust absorption/no/adjust self scatter to force ks balance

...edit position names...

card
position scheme position
1 chi
2 nucgeom 3
3 total
4 angle
5 edit1

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

<table>
<thead>
<tr>
<th>isotope number</th>
<th>name</th>
<th>order</th>
<th>header card</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>is01</td>
<td>p0</td>
<td>core</td>
</tr>
<tr>
<td>2</td>
<td>is02</td>
<td>p0</td>
<td>eclude</td>
</tr>
<tr>
<td>3</td>
<td>is03</td>
<td>p0</td>
<td>macrol</td>
</tr>
<tr>
<td>4</td>
<td>is04</td>
<td>p0</td>
<td>eontrol</td>
</tr>
<tr>
<td>5</td>
<td>is05</td>
<td>p0</td>
<td>radalk</td>
</tr>
</tbody>
</table>

*Key end card like read

*Key end block iv read-mac

THREEDANT USER’S GUIDE Version 3.0 6-79
**APPENDIX A: SAMPLE INPUT**

Sample Problem: Output Listing

```
*                                                                                       *
*   * mixing instructions ...                                                           *
*                                                                                       *
*   **mix   comp  density  comp  density etc.**                                          *
*                                                                                       *
*   mats                                               
*   1. isoc  isoc  1.000000E+00,                                                    *
*   2. isoc  isoc  1.000000E+00,                                                    *
*   3. isoc  isoc  1.000000E+00,                                                    *
*   4. isoc  isoc  1.000000E+00,                                                    *
*   5. isoc  isoc  1.000000E+00,                                                    *
*                                                                                       *
* **Key start mix card as:**                                                 
* **Key end mix card as:**                                                          
* **Key end block v read-solver**                                                 
* **Key end block v1 read-edit**                                                  
* **Key end input module**                                                          
```
Sample Problem: Output Listing

---

...block v — solver input...

raw input defaulted

...required input (array name = solin)...

1 1 ievt 0/2/3/4 - 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 iact legends order of scattering
0 1 ich 0/1 - direct/adjoint - mode of calculation (default-direct)
         1 left boundary condition
         2 vacuum/reflective
0 0 ib 0/1/3 - right boundary condition
         1 vacuum/reflective
0 0 ib 0/1/3 - top boundary condition
         1 vacuum/reflective
1 1 ib 0/1/3 - bottom boundary condition
         1 vacuum/reflective
0 0 ihk 0/1/3 - back boundary condition
         1 vacuum/reflective
0 0 ibhr 0/1/3 - front boundary condition
         1 vacuum/reflective

...convergence controls (array name = iter)...

5.00E-05 5.00E-05 evpi inner iteration convergence criterion (default=0.001)
1 1 illi maximum number of inner iterations per group until fission sources 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 icat maximum number of outer iterations (default=20)
0 0 ict time iteration time limit (seconds)

---

...block v — solver input (continued)...

raw input defaulted

...miscellaneous parameters (array name = misc)...

1.000E+00 1.000E+00 norm normalization factor
0 0 no сит fissions zero when source calculation (0/1 no/yes)
0 0 influx 0/1 no/yes - read input flux from file influx (influx for adjoint)
0 0 iusers 0/1 no/yes - read input sources from file users
0 0 nodal 0/1 no/3/3 nodal option for zxy
0 1 iquad -3/-2/1/2/3 - source quadrature constants (default=1)
         -3 photon file
         -2 hybrid product set (triangular arrangement)
         1 old bottom build-in set
         2 product set (rectangular arrangement)
         3 card input

...output controls (array name = solout)...

0 influx 0/1/2 none/transport/none - flux print
2 escout 0/1/2 none/principal/all - macroscopic cross section print
0 esca 0/1/2 yes - print final fission source rate
0 scrs 0/1/2 no/yes read/normalized/no - print inhomogeneous source
0 spp 0/1/2 no/yes - print angular fluxes
0 mflux 0/1/2 no/yes - write angular fluxes to file mflux or mflux(if ich=1)
0 0 mflux 0/1/2 yes - write file mfluxes to file mflux
0 df 0/1/2 yes - print course mesh balance

...parameters inferred from input arrays...

1 ichi 0/1/2 none/one-echi canone ech7
0 ishef 0/1/2 none/one density vector/fall matrix
0 ishef 0/1/2 yes - use y density vector
0 ishef source anisotropy
0 ishef number of source moments input
0 ishef number of source moments output
0 lsoy number of source moments input
0 lsdy number of source moments output
0 ishef number of source moments input
0 lsdy number of source moments output
0 ishef -1/-0/1/2 isotropic/none/all angles/none - top boundary source
0 ishef -1/-0/1/2 isotropic/none/all angles/none - bottom boundary source
0 ishef -1/-0/1/2 isotropic/none/all angles/none - front boundary source
0 ishef -1/-0/1/2 isotropic/none/all angles/none - back boundary source

---

THREEDANT USER'S GUIDE  Version 3.0  6-81
...parameters from block i...

14 igem 14/15 x-y-z/4-theta
4 rgroup number of energy groups
8 lam angular quadrature order
5 sm number of permanent materials
5 rzone number of zones
8 is number of coarse mesh x intervals
8 js number of coarse mesh y intervals
4 ks number of coarse mesh x intervals
14 it number of fine mesh x intervals
14 jk number of fine mesh y intervals
30 kc number of fine mesh x intervals
Sample Problem: Output Listing

...material assignments to zones...

*key start mats to zones*

<table>
<thead>
<tr>
<th>zone</th>
<th>material no.</th>
<th>name</th>
<th>no.</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mat1</td>
<td>1</td>
<td>mat1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>mat2</td>
<td>2</td>
<td>mat2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>mat3</td>
<td>3</td>
<td>mat3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>mat4</td>
<td>4</td>
<td>mat4</td>
<td>4</td>
</tr>
</tbody>
</table>

*key start storage map*

**scm storage summary...**

- total scm required for this problem: 90713
- maximum scm available (maxarea): 300000
- scm required for transport: 69790
- scm required for diffusion: 90668

**ion storage summary...**

- ion selected for this problem: 203602
- maximum ion specified (maxcen): 600000

- storage required for all quantities in core: 203602
- storage required for all quantities on disk: 73162

203602 words ion required to add

**zero flux**

**key start ex constants**

- 8 constants for groups 1 to 4

<table>
<thead>
<tr>
<th>m</th>
<th>era</th>
<th>xi</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.19232747E+00</td>
<td>0.96229948E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.77735072E+00</td>
<td>0.79352178E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.19232747E+00</td>
<td>0.79352178E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>4</td>
<td>0.77735072E+00</td>
<td>0.79352178E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>5</td>
<td>0.77735072E+00</td>
<td>0.79352178E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>6</td>
<td>0.19232747E+00</td>
<td>0.79352178E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>7</td>
<td>0.96229948E+00</td>
<td>0.19232747E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>8</td>
<td>0.79352178E+00</td>
<td>0.19232747E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>9</td>
<td>0.77735072E+00</td>
<td>0.19232747E+00</td>
<td>0.19232747E+00</td>
</tr>
<tr>
<td>10</td>
<td>0.19232747E+00</td>
<td>0.96229948E+00</td>
<td>0.19232747E+00</td>
</tr>
</tbody>
</table>
Three-dimensional coarse mesh geometry edit...

```

    n  xmesh  xint  deltaz  ymesh  yint  deltay  zmesh  zint  deltax
  0  0.00E+00  0.00E+00  0.00E+00
  1  1.50E+01  3.00E+00  5.00E+00  1.50E+01  2.00E+01  4.50E+00
  2  3.00E+01  3.50E+00  1.50E+01  2.00E+01  4.50E+00  7.50E+00
  3  3.50E+01  1.00E+00  3.00E+01  3.00E+01  1.00E+00  4.50E+00
  4  4.00E+01  1.00E+00  4.00E+01  2.00E+01  4.50E+00  1.50E+00
  5  4.50E+01  1.00E+00  4.50E+01  1.50E+00  4.50E+00
  6  5.00E+01  1.00E+00  5.00E+01  1.50E+00  4.50E+00
  7  5.50E+01  1.00E+00  5.50E+01  1.50E+00  4.50E+00
  8  7.00E+01  3.00E+00  7.00E+01  3.00E+00  4.50E+00
```
```
<table>
<thead>
<tr>
<th>Zone</th>
<th>CHI</th>
<th>Neutron Cross Section</th>
<th>Total Absorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3.8052E-01</td>
<td>6.2348E-03</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3.7513E-01</td>
<td>8.8962E-03</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1.8036E-01</td>
<td>3.5448E-04</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3.6056E-01</td>
<td>2.9843E-02</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3.4841E-01</td>
<td>9.5703E-03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zone</th>
<th>CHI</th>
<th>Neutron Cross Section</th>
<th>Total Absorption</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3.8052E-01</td>
<td>6.2348E-03</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3.7513E-01</td>
<td>8.8962E-03</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1.8036E-01</td>
<td>3.5448E-04</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3.6056E-01</td>
<td>2.9843E-02</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3.4841E-01</td>
<td>9.5703E-03</td>
</tr>
</tbody>
</table>
```
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>4</td>
<td>3.7489E-01</td>
<td>1.6955E-03</td>
<td>6.6839E-07</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4</td>
<td>2.0888E-01</td>
<td>1.0126E-03</td>
<td>1.7766E-07</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
<td>6.1870E-01</td>
<td>7.6603E-03</td>
<td>8.7214E-07</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>3.0343E-01</td>
<td>1.2703E-03</td>
<td>1.9978E-07</td>
</tr>
</tbody>
</table>

---

**Sample Problem: Output Listing**

---

THREEDANT USER’S GUIDE  Version 3.0

6-87
### ...iteration controls and criteria...

#### ***iteration criteria***

<table>
<thead>
<tr>
<th>criterion</th>
<th>quantity to test</th>
<th>value</th>
<th>action taken if value exceeded</th>
</tr>
</thead>
<tbody>
<tr>
<td>itul</td>
<td>inner iteration count until near lambda (i.e., fissile source) convergence</td>
<td>1</td>
<td>terminates inner</td>
</tr>
<tr>
<td>iton</td>
<td>inner iteration count when near lambda (i.e., fissile source) convergence</td>
<td>30</td>
<td>terminates inner</td>
</tr>
<tr>
<td>epsi</td>
<td>fractional fission flux change per inner</td>
<td>5.00E-05</td>
<td>does another inner</td>
</tr>
<tr>
<td></td>
<td>diffusion sub-inners</td>
<td></td>
<td></td>
</tr>
<tr>
<td>crits</td>
<td>sub-inner iteration count</td>
<td>22</td>
<td>terminates sub-inners</td>
</tr>
<tr>
<td>epsi</td>
<td>fractional fission flux change per sub-inner (see note below)</td>
<td>5.00E-05</td>
<td>does another sub-inner</td>
</tr>
<tr>
<td></td>
<td>final convergence criteria</td>
<td></td>
<td></td>
</tr>
<tr>
<td>crits</td>
<td>outer iteration count</td>
<td>11</td>
<td>quits with error message</td>
</tr>
<tr>
<td>epsi</td>
<td>transport lambda-1.0 (see note below)</td>
<td>5.00E-05</td>
<td>does another outer</td>
</tr>
</tbody>
</table>

*Note: epsi, 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.*

#### ***flux and eigenvalue convergence as monitored by threads***

*Key word iteration monitor*

<table>
<thead>
<tr>
<th>crits name</th>
<th>diffusion</th>
<th>k-eff</th>
<th>eigenvalue</th>
<th>lambda-1</th>
<th>flux change</th>
<th>flux change</th>
<th>flux change converged</th>
</tr>
</thead>
<tbody>
<tr>
<td>time outer</td>
<td>diff</td>
<td>k-eff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(sec)</td>
<td>(sec)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.41</td>
<td>0 1 0 0 0</td>
<td>0.9451561</td>
<td>8.53213E-03</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>5.76</td>
<td>1 0 0 0 0</td>
<td>0.9451561</td>
<td>8.53213E-03</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>6.88</td>
<td>2 4 0 0 0</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>2.39734E-01</td>
<td>2.39734E-01</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>7.54</td>
<td>3 4 1 0 0</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>2.39734E-01</td>
<td>2.39734E-01</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>9.27</td>
<td>3 4 1 0 0</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>2.39734E-01</td>
<td>2.39734E-01</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>10.76</td>
<td>4 4 1 0 0</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>2.39734E-01</td>
<td>2.39734E-01</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>12.25</td>
<td>5 4 1 0 0</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>2.39734E-01</td>
<td>2.39734E-01</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>13.74</td>
<td>6 4 1 0 0</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>2.39734E-01</td>
<td>2.39734E-01</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

---

**inner iteration summary for outer iteration no. 7**

<table>
<thead>
<tr>
<th>crits name</th>
<th>diffusion</th>
<th>k-eff</th>
<th>eigenvalue</th>
<th>lambda-1</th>
<th>flux change</th>
<th>flux change</th>
<th>flux change converged</th>
</tr>
</thead>
<tbody>
<tr>
<td>time outer</td>
<td>diff</td>
<td>k-eff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(sec)</td>
<td>(sec)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.52</td>
<td>7 9 1 1 1</td>
<td>0.9734710</td>
<td>2.99400E-02</td>
<td>4.59971E-05</td>
<td>5.09646E-06</td>
<td>yes</td>
<td></td>
</tr>
</tbody>
</table>

SSSSSS all convergence criteria satisfied SSSSSS

particle balance = 3.813378E-08 total inner all outer = 29
### THREEDANT USER'S GUIDE

**APPENDIX A: SAMPLE INPUT**

Sample Problem: Output Listing

---

```
...output section...
...title-- 3D model of a small FR...
...system balance tables... (neutrons only)

```

#### Key start balance table

<table>
<thead>
<tr>
<th>gp</th>
<th>source</th>
<th>fission source</th>
<th>absorption</th>
<th>in scatter</th>
<th>self scatter</th>
<th>out scatter</th>
<th>net leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000000E+00</td>
<td>5.8319000E-01</td>
<td>9.3606435E-02</td>
<td>-2.5934810E-03</td>
<td>9.0152126E-01</td>
<td>4.7787527E-01</td>
<td>1.1873786E-02</td>
</tr>
<tr>
<td>2</td>
<td>0.0000000E+00</td>
<td>4.0560000E-00</td>
<td>2.4701782E+00</td>
<td>4.5277070E-01</td>
<td>1.5724665E-01</td>
<td>5.3359048E-01</td>
<td>7.7843956E-02</td>
</tr>
<tr>
<td>3</td>
<td>0.0000000E+00</td>
<td>1.2300000E-00</td>
<td>4.3396872E-01</td>
<td>5.8368621E-01</td>
<td>1.9305246E-01</td>
<td>6.5303069E-02</td>
<td>7.0817732E-02</td>
</tr>
<tr>
<td>4</td>
<td>0.0000000E+00</td>
<td>0.0000000E+00</td>
<td>6.0224756E-02</td>
<td>6.3357645E-02</td>
<td>9.8799113E-02</td>
<td>1.2076664E-02</td>
<td>5.1316768E-03</td>
</tr>
<tr>
<td>tot</td>
<td>0.0000000E+00</td>
<td>1.0000000E+00</td>
<td>8.3423560E-01</td>
<td>1.0764970E+00</td>
<td>3.6916735E+00</td>
<td>1.0765663E+00</td>
<td>1.6560468E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gp</th>
<th>source</th>
<th>fission source</th>
<th>top leakage</th>
<th>horizontal leakage</th>
<th>bottom leakage</th>
<th>back leakage</th>
<th>f-back leakage particle balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.9897362E-03</td>
<td>3.9897362E-03</td>
<td>4.1657121E-03</td>
<td>4.1657121E-03</td>
<td>1.8428040E-03</td>
<td>1.6956011E-03</td>
<td>3.9000586E-03</td>
</tr>
<tr>
<td>2</td>
<td>2.5613078E-02</td>
<td>2.5613078E-02</td>
<td>2.6512387E-02</td>
<td>2.6512387E-02</td>
<td>1.2852936E-02</td>
<td>1.2852936E-02</td>
<td>5.4039123E-02</td>
</tr>
<tr>
<td>3</td>
<td>2.3914820E-02</td>
<td>2.3914820E-02</td>
<td>2.4462588E-02</td>
<td>2.4462588E-02</td>
<td>1.1253132E-02</td>
<td>1.1253132E-02</td>
<td>3.1154973E-02</td>
</tr>
<tr>
<td>4</td>
<td>1.5625989E+00</td>
<td>1.5625989E+00</td>
<td>1.6084963E+00</td>
<td>1.6084963E+00</td>
<td>7.3186476E-02</td>
<td>7.3186476E-02</td>
<td>4.2588296E-02</td>
</tr>
<tr>
<td>tot</td>
<td>5.5096326E-02</td>
<td>5.5096326E-02</td>
<td>5.6741304E-02</td>
<td>5.6741304E-02</td>
<td>2.6928566E-02</td>
<td>2.6928566E-02</td>
<td>3.8133656E-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gp</th>
<th>source</th>
<th>fission source</th>
<th>left leakage</th>
<th>bottom leakage</th>
<th>front leakage</th>
<th>prod spectrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.6605245E-16</td>
<td>-4.7144747E-16</td>
<td>1.8427791E-03</td>
<td>2.6283412E-01</td>
<td>3.2640597E-01</td>
<td>1.0000000E+00</td>
</tr>
<tr>
<td>2</td>
<td>9.9220071E-16</td>
<td>-2.7755768E-15</td>
<td>1.2852936E-02</td>
<td>1.2852936E-02</td>
<td>7.3186476E-02</td>
<td>2.3914820E-02</td>
</tr>
<tr>
<td>3</td>
<td>7.7731625E-16</td>
<td>-2.4324917E-15</td>
<td>1.1253132E-02</td>
<td>1.1253132E-02</td>
<td>4.2588296E-02</td>
<td>2.5613078E-02</td>
</tr>
<tr>
<td>4</td>
<td>3.4694476E-17</td>
<td>-1.2877888E-15</td>
<td>9.7318478E-04</td>
<td>4.3835442E-02</td>
<td>2.6928566E-02</td>
<td>5.5096326E-02</td>
</tr>
<tr>
<td>tot</td>
<td>1.9775848E-15</td>
<td>-5.0286709E-15</td>
<td>2.6928566E-02</td>
<td>2.6928566E-02</td>
<td>4.2588296E-02</td>
<td>5.5096326E-02</td>
</tr>
</tbody>
</table>

---

**Multigrid work units...**

```
Multigrid average convergence rate by group...
```

```
By group...
1 242.27 2 245.49 3 236.05 4 235.39
```

**integral survey information**

```
summary integral-eff 9.7347913E-03
integral-fission-i neutron 0.0000000E+00
integral-fission-n neutron 1.0000000E+00
integral-absorption-i neutron 6.3423565E-01
integral-in-scatter-i neutron 1.0764970E-04
integral-self-scatter-i neutron 3.6917735E+01
integral-out-scatter-i neutron 1.0764970E-04
integral-mass leakage-i neutron 1.8427791E-03
integral-right leakage-i neutron 5.2056256E-02
integral-horizontal leakage-i neutron 5.2056256E-02
integral-top leakage-i neutron 5.6741304E-02
integral-vertical leakage-i neutron 5.6741304E-02
integral-back leakage-i neutron 2.6928566E-02
integral-fr-back leakage-i neutron 5.3870652E-02
integral-particle hal-i neutron 1.1530471E-07
```

```
...interface file rflux written...
...interface file sums written...
```

**threednt: iteration time, mins 2.76248E-01**

---

THREEDANT USER'S GUIDE  Version 3.0 6-89
*May start response functions*

---

```
point  av pd  mpfl
2745  2.5000  5.20 9289-05
2746  7.5000  5.06519-05
2747  12.5000 8.79278-05
2748  17.5000 8.12118-05
2749  22.5000 7.71118-05
2750  27.5000 6.97132-05
2751  32.5000 5.85823-05
2752  37.5000 4.20562-05
2753  42.5000 3.86194-05
2754  47.5000 3.53875-05
2755  52.5000 2.60786-05
2756  57.5000 1.29354-05
2757  62.5000 6.71218-06
2758  67.5000 3.47125-06
2759  72.5000 9.07354-06
2760  77.5000 8.62228-05
2761  82.5000 8.44229-05
2801  2.5000 7.95213-05
2802  2.5000 7.33235-05
2829  2.5000 6.62768-05
2830  2.5000 5.823999-05
2857  2.5000 4.924859-05
2871  2.5000 3.96686-05
2885  2.5000 2.76707-05
2899  2.5000 1.45694-05
2913  2.5000 7.15846-05
2927  2.5000 3.713389-06
```

---

```
zone  volume  mpfl
1  2.64000e+06  1.00369e+05
2  3.80000e+04  4.49845e+04
3  3.75000e+03  9.32943e+02
4  6.75000e+03  1.57966e+03
```

---

```
point  av pd  mpfl
2745  2.5000  5.06399-04
2746  7.5000  4.89483-04
2747  12.5000 4.81978-04
2748  17.5000 4.57498-04
2749  22.5000 4.25018-04
2750  27.5000 3.84812-04
2751  32.5000 3.13900-04
2752  37.5000 2.64049-04
2753  42.5000 2.30092-04
2754  47.5000 2.13018-04
2755  52.5000 1.63176-04
2756  57.5000 1.08919-04
2757  62.5000 6.43279-05
2758  67.5000 3.01813-05
2759  2.5000 4.96589-04
2773  2.5000 4.98939-04
2787  2.5000 4.57718-04
2801  2.5000 4.37238-04
2823  2.5000 4.03825-04
2852  2.5000 3.647999-04
2843  2.5000 3.21000-04
2857  2.5000 3.71484-04
2871  2.5000 2.22573-04
2895  2.5000 1.67910-04
2899  2.5000 1.14813-04
2913  2.5000 6.67808-05
2927  2.5000 3.14302-05
```
### APPENDIX A: SAMPLE INPUT

#### Sample Problem: Output Listing

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>mpfli</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6400E+06</td>
<td>6.18625E-01</td>
</tr>
<tr>
<td>2</td>
<td>9.6000E+04</td>
<td>4.43925E-00</td>
</tr>
<tr>
<td>3</td>
<td>3.7500E+03</td>
<td>6.09456E-01</td>
</tr>
<tr>
<td>4</td>
<td>3.7500E+03</td>
<td>6.09456E-01</td>
</tr>
<tr>
<td>5</td>
<td>3.6750E+05</td>
<td>1.1571E+01</td>
</tr>
<tr>
<td>sum</td>
<td>7.3500E+05</td>
<td>7.86659E+01</td>
</tr>
</tbody>
</table>

...point edit for broad group 3 ...

#### response functions ...

<table>
<thead>
<tr>
<th>point</th>
<th>av rad</th>
<th>mpfli</th>
</tr>
</thead>
<tbody>
<tr>
<td>2745</td>
<td>2.5000</td>
<td>3.24572E-04</td>
</tr>
<tr>
<td>2746</td>
<td>7.5000</td>
<td>3.94389E-04</td>
</tr>
<tr>
<td>2747</td>
<td>12.5000</td>
<td>3.93865E-04</td>
</tr>
<tr>
<td>2748</td>
<td>17.5000</td>
<td>2.94538E-04</td>
</tr>
<tr>
<td>2749</td>
<td>22.5000</td>
<td>2.76504E-04</td>
</tr>
<tr>
<td>2750</td>
<td>27.5000</td>
<td>2.53853E-04</td>
</tr>
<tr>
<td>2751</td>
<td>32.5000</td>
<td>2.76659E-04</td>
</tr>
<tr>
<td>2752</td>
<td>37.5000</td>
<td>2.05403E-04</td>
</tr>
<tr>
<td>2753</td>
<td>42.5000</td>
<td>1.84316E-04</td>
</tr>
<tr>
<td>2754</td>
<td>47.5000</td>
<td>1.63344E-04</td>
</tr>
<tr>
<td>2755</td>
<td>52.5000</td>
<td>1.44649E-04</td>
</tr>
<tr>
<td>2756</td>
<td>57.5000</td>
<td>9.60515E-05</td>
</tr>
<tr>
<td>2757</td>
<td>62.5000</td>
<td>6.47537E-05</td>
</tr>
<tr>
<td>2758</td>
<td>67.5000</td>
<td>2.99722E-05</td>
</tr>
<tr>
<td>2759</td>
<td>72.5000</td>
<td>3.93258E-04</td>
</tr>
<tr>
<td>2760</td>
<td>77.5000</td>
<td>1.11792E-04</td>
</tr>
<tr>
<td>2761</td>
<td>82.5000</td>
<td>2.98044E-04</td>
</tr>
<tr>
<td>2762</td>
<td>87.5000</td>
<td>2.60164E-04</td>
</tr>
<tr>
<td>2763</td>
<td>92.5000</td>
<td>1.20294E-04</td>
</tr>
<tr>
<td>2764</td>
<td>97.5000</td>
<td>9.83766E-05</td>
</tr>
<tr>
<td>2765</td>
<td>1.025000</td>
<td>1.53428E-04</td>
</tr>
<tr>
<td>2766</td>
<td>1.075000</td>
<td>1.26294E-04</td>
</tr>
</tbody>
</table>

...zone edit for broad group 3 ...

#### response functions ...

<table>
<thead>
<tr>
<th>zone</th>
<th>volume</th>
<th>mpfli</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6400E+06</td>
<td>6.18625E-01</td>
</tr>
<tr>
<td>2</td>
<td>9.6000E+04</td>
<td>4.43925E-00</td>
</tr>
<tr>
<td>3</td>
<td>3.7500E+03</td>
<td>6.09456E-01</td>
</tr>
<tr>
<td>4</td>
<td>3.7500E+03</td>
<td>6.09456E-01</td>
</tr>
<tr>
<td>5</td>
<td>3.6750E+05</td>
<td>1.1571E+01</td>
</tr>
<tr>
<td>sum</td>
<td>7.3500E+05</td>
<td>7.86659E+01</td>
</tr>
</tbody>
</table>

...point edit for broad group 4 ...

#### response functions ...

<table>
<thead>
<tr>
<th>point</th>
<th>av rad</th>
<th>mpfli</th>
</tr>
</thead>
<tbody>
<tr>
<td>2745</td>
<td>2.5000</td>
<td>1.11766E-05</td>
</tr>
<tr>
<td>2746</td>
<td>7.5000</td>
<td>1.03052E-05</td>
</tr>
<tr>
<td>2747</td>
<td>12.5000</td>
<td>1.06839E-05</td>
</tr>
<tr>
<td>2748</td>
<td>17.5000</td>
<td>1.02556E-05</td>
</tr>
<tr>
<td>2749</td>
<td>22.5000</td>
<td>9.67713E-06</td>
</tr>
<tr>
<td>2750</td>
<td>27.5000</td>
<td>9.14910E-06</td>
</tr>
<tr>
<td>2751</td>
<td>32.5000</td>
<td>9.31938E-06</td>
</tr>
<tr>
<td>2752</td>
<td>37.5000</td>
<td>1.00037E-05</td>
</tr>
<tr>
<td>2753</td>
<td>42.5000</td>
<td>9.36134E-06</td>
</tr>
<tr>
<td>2754</td>
<td>47.5000</td>
<td>7.39846E-06</td>
</tr>
<tr>
<td>2755</td>
<td>52.5000</td>
<td>5.55638E-06</td>
</tr>
<tr>
<td>2756</td>
<td>57.5000</td>
<td>4.52485E-06</td>
</tr>
<tr>
<td>2757</td>
<td>62.5000</td>
<td>3.73016E-06</td>
</tr>
<tr>
<td>2758</td>
<td>67.5000</td>
<td>2.90281E-06</td>
</tr>
<tr>
<td>2759</td>
<td>72.5000</td>
<td>1.02588E-05</td>
</tr>
<tr>
<td>2760</td>
<td>77.5000</td>
<td>1.07122E-05</td>
</tr>
<tr>
<td>2761</td>
<td>82.5000</td>
<td>1.02707E-05</td>
</tr>
<tr>
<td>2762</td>
<td>87.5000</td>
<td>6.69486E-06</td>
</tr>
<tr>
<td>2763</td>
<td>92.5000</td>
<td>6.48889E-06</td>
</tr>
<tr>
<td>2764</td>
<td>97.5000</td>
<td>6.20081E-06</td>
</tr>
<tr>
<td>2765</td>
<td>1.025000</td>
<td>7.34315E-06</td>
</tr>
<tr>
<td>2766</td>
<td>1.075000</td>
<td>6.48765E-06</td>
</tr>
<tr>
<td>2767</td>
<td>1.125000</td>
<td>5.69635E-06</td>
</tr>
<tr>
<td>2768</td>
<td>1.175000</td>
<td>5.21662E-06</td>
</tr>
<tr>
<td>2769</td>
<td>1.225000</td>
<td>4.84458E-06</td>
</tr>
<tr>
<td>2770</td>
<td>1.275000</td>
<td>4.51729E-06</td>
</tr>
<tr>
<td>2771</td>
<td>1.325000</td>
<td>4.12836E-06</td>
</tr>
<tr>
<td>2772</td>
<td>1.375000</td>
<td>3.75228E-06</td>
</tr>
</tbody>
</table>

...zone edit for broad group 4 ...

#### response functions ...

---

**THREEDANT USER'S GUIDE** Version 3.0

Page 6-93
### APPENDIX A: SAMPLE INPUT

#### Sample Problem: Output Listing

<table>
<thead>
<tr>
<th>zone</th>
<th>value</th>
<th>mpfl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6400E-05</td>
<td>1.59645E+00</td>
</tr>
<tr>
<td>2</td>
<td>9.6000E-04</td>
<td>3.30324E+00</td>
</tr>
<tr>
<td>3</td>
<td>3.7500E+03</td>
<td>2.25313E+02</td>
</tr>
<tr>
<td>4</td>
<td>3.7500E+03</td>
<td>2.25313E+02</td>
</tr>
<tr>
<td>5</td>
<td>1.6750E-05</td>
<td>7.35933E+01</td>
</tr>
</tbody>
</table>

**sum**: 7.3500E+05 2.73325E+00

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

#### response functions

<table>
<thead>
<tr>
<th>point</th>
<th>av rad</th>
<th>mpfl</th>
</tr>
</thead>
<tbody>
<tr>
<td>2745</td>
<td>2.5000</td>
<td>9.34516E-04</td>
</tr>
<tr>
<td>2746</td>
<td>7.5000</td>
<td>9.13468E-04</td>
</tr>
<tr>
<td>2747</td>
<td>12.5000</td>
<td>8.89459E-04</td>
</tr>
<tr>
<td>2748</td>
<td>17.5000</td>
<td>8.40422E-04</td>
</tr>
<tr>
<td>2749</td>
<td>22.5000</td>
<td>7.93126E-04</td>
</tr>
<tr>
<td>2750</td>
<td>27.5000</td>
<td>7.10526E-04</td>
</tr>
<tr>
<td>2751</td>
<td>32.5000</td>
<td>6.29518E-04</td>
</tr>
<tr>
<td>2752</td>
<td>37.5000</td>
<td>5.49548E-04</td>
</tr>
<tr>
<td>2753</td>
<td>42.5000</td>
<td>4.69632E-04</td>
</tr>
<tr>
<td>2754</td>
<td>47.5000</td>
<td>4.00292E-04</td>
</tr>
<tr>
<td>2755</td>
<td>52.5000</td>
<td>3.17406E-04</td>
</tr>
<tr>
<td>2756</td>
<td>57.5000</td>
<td>2.24848E-04</td>
</tr>
<tr>
<td>2757</td>
<td>62.5000</td>
<td>1.40125E-04</td>
</tr>
<tr>
<td>2758</td>
<td>67.5000</td>
<td>6.54768E-05</td>
</tr>
<tr>
<td>2759</td>
<td>2.5000</td>
<td>9.21452E-04</td>
</tr>
<tr>
<td>2773</td>
<td>2.5000</td>
<td>8.95710E-04</td>
</tr>
<tr>
<td>2777</td>
<td>2.5000</td>
<td>8.57149E-04</td>
</tr>
<tr>
<td>2801</td>
<td>2.5000</td>
<td>8.07464E-04</td>
</tr>
<tr>
<td>2813</td>
<td>2.5000</td>
<td>7.46135E-04</td>
</tr>
<tr>
<td>2829</td>
<td>2.5000</td>
<td>6.79562E-04</td>
</tr>
<tr>
<td>2943</td>
<td>2.5000</td>
<td>5.97548E-04</td>
</tr>
<tr>
<td>2957</td>
<td>2.5000</td>
<td>5.10999E-04</td>
</tr>
<tr>
<td>2871</td>
<td>2.5000</td>
<td>4.23211E-04</td>
</tr>
<tr>
<td>2885</td>
<td>2.5000</td>
<td>3.29258E-04</td>
</tr>
<tr>
<td>2899</td>
<td>2.5000</td>
<td>2.33093E-04</td>
</tr>
<tr>
<td>2913</td>
<td>2.5000</td>
<td>1.45997E-04</td>
</tr>
<tr>
<td>2927</td>
<td>2.5000</td>
<td>8.42892E-05</td>
</tr>
</tbody>
</table>

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

#### response functions

<table>
<thead>
<tr>
<th>zone</th>
<th>value</th>
<th>mpfl</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.6400E-05</td>
<td>1.59629E+02</td>
</tr>
<tr>
<td>2</td>
<td>9.6000E-04</td>
<td>3.30346E+00</td>
</tr>
<tr>
<td>3</td>
<td>3.7500E+03</td>
<td>2.25315E+02</td>
</tr>
<tr>
<td>4</td>
<td>3.7500E+03</td>
<td>2.25315E+02</td>
</tr>
<tr>
<td>5</td>
<td>1.6750E-05</td>
<td>7.35932E+01</td>
</tr>
</tbody>
</table>

**sum**: 7.3500E+05 1.53621E+02
### Run Highlights

All modules are tentatively go.

Interface file paged written.

Cross sections from cards.

Word overidden for high.

Interface mixing files written.

Interface file augmt written.

$ files written.

Interface file update written.

Interface file edit written.

Start solve execution.

All convergence criteria met.

Interface file flux written.

Interface file mens written.

Start edit execution.

Edits completed.

### Storage and Timing History

<table>
<thead>
<tr>
<th>Module</th>
<th>Words</th>
<th>Limit</th>
<th>Words</th>
<th>Limit</th>
<th>Secs</th>
<th>Secs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16.8</td>
<td>1.4</td>
</tr>
<tr>
<td>100</td>
<td>9548</td>
<td>30000</td>
<td>0</td>
<td>0</td>
<td>4.3</td>
<td>0.2</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td>0.0</td>
</tr>
<tr>
<td>102</td>
<td>1135</td>
<td>30000</td>
<td>289</td>
<td>60000</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>103</td>
<td>72</td>
<td>30000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>104</td>
<td>2077</td>
<td>30000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>126</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>106</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>107</td>
<td>215</td>
<td>30000</td>
<td>190</td>
<td>60000</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>108</td>
<td>1068</td>
<td>30000</td>
<td>6</td>
<td>60000</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>109</td>
<td>15442</td>
<td>30000</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>112</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>200</td>
<td>90719</td>
<td>30000</td>
<td>20360</td>
<td>60000</td>
<td>12.2</td>
<td>1.0</td>
</tr>
<tr>
<td>201</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>203</td>
<td>1393</td>
<td>30000</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>204</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>226</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>206</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>207</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.8</td>
<td>0.6</td>
</tr>
<tr>
<td>208</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>210</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.1</td>
</tr>
<tr>
<td>211</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>300</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>301</td>
<td>36090</td>
<td>30000</td>
<td>123</td>
<td>60000</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>302</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

...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

```bash
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

<table>
<thead>
<tr>
<th>LIB</th>
<th>SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACRXS</td>
<td>Current Working Directory (CWD).</td>
</tr>
<tr>
<td>GRUPXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>ISOTXS</td>
<td>SNXSPATH, then CWD.</td>
</tr>
<tr>
<td>BXSLIB</td>
<td>SNXSPATH, then CWD, but see text below.</td>
</tr>
<tr>
<td>ODNINP</td>
<td>None, the library is contained in the input file.</td>
</tr>
<tr>
<td>MACBCD</td>
<td>CWD</td>
</tr>
<tr>
<td>XSLIBB</td>
<td>CWD</td>
</tr>
<tr>
<td>MENDF&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Path defined in the code on UNICOS. MENDF binaries are unavailable for SUN.</td>
</tr>
<tr>
<td>MENDFG&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>XSLIB</td>
<td>SNXSPATH, then CWD</td>
</tr>
<tr>
<td>other</td>
<td>For any name other than those above, the program will assume the form is XSLIB and search for it in SNXS-PATH, then CWD.</td>
</tr>
</tbody>
</table>

---

<sup>a</sup> Available only at Los Alamos.  
<sup>b</sup> 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
# TABLE OF CONTENTS

**TABLE OF CONTENTS** ................................................................. 7-3

**LIST OF FIGURES** ........................................................................ 7-5

**LIST OF TABLES** .......................................................................... 7-7

**INTRODUCTION** ............................................................................ 7-9

**MORE DETAILS ON BLOCK-I INPUT** ........................................... 7-11
  - Angular Quadrature (ISN) .......................................................... 7-11
  - Geometry (NZONE, IM, IT) ......................................................... 7-11
  - MAXSCM, MAXLCM ..................................................................... 7-11
  - Execution/File Suppression Flags ............................................... 7-12

**MORE DETAILS ON GEOMETRY INPUT** ...................................... 7-13

**MORE DETAILS ON SOLVER INPUT** .......................................... 7-15
  - Iteration Strategy ....................................................................... 7-15
  - Convergence Criteria .................................................................. 7-19
    - Inner Iteration Convergence ................................................... 7-19
    - Diffusion Sub-Outer Iteration Convergence ............................ 7-19
    - Transport Source Iteration Convergence ............................... 7-20
    - Final Convergence .................................................................. 7-20
  - Grey Acceleration of Upscatter .................................................. 7-21
  - Iteration Monitor Print ............................................................... 7-21
    - General Aspects of the Monitor Print .................................... 7-21
    - Warning Messages and Their Meanings .................................. 7-22
  - Boundary Conditions ................................................................... 7-23
  - Input of Quadrature Sets ............................................................ 7-24
  - Zone-Dependent Fission Fractions (the CHI Array) .................... 7-25
  - Input of Inhomogeneous Sources .............................................. 7-26
    - Distributed Source Input ....................................................... 7-26
      - Surface (Boundary) Source Input ....................................... 7-28
  - Normalization of the Calculation (the NORM Parameter) ............ 7-30
  - Transport Corrections for the Cross Sections (TRCOR) .............. 7-31
  - Buckling Corrections .................................................................. 7-33
  - Eigenvalue Searches ................................................................... 7-33
  - Adjoint Computations .................................................................. 7-37

**REFERENCES** ................................................................................ 7-39
LIST OF FIGURES

Figure 7.1: Simplified flow diagram of SOLVER iteration strategy. ...................... 7-18
Figure 7.2: Ordering in slab geometry. ................................................................. 7-28
Figure 7.3: Quadrature points in cylindrical geometry. ........................................... 7-30
Figure 7.4: Variation of $\lambda$ during a hypothetical eigenvalue search. .................. 7-37
LIST OF TABLES

Table 7.1:  Source Ordering Index in Slab Geometry ............................................. 7-29
Table 7.2:  Source Ordering Index in Cylindrical Geometry ............................... 7-30
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.
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.
Geometry-related information is passed to the appropriate SOLVER module and the EDIT module solely by means of the GEODST standard interface file. 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 "ONDANT, 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 ≤ n ≤ 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.
Iteration Strategy

As described in "Iteration Procedure and Diffusion Synthetic Acceleration" on page 12-14 of this document, the ONEDANT, TWODANT, TWODANT/GQ, and THREE-DANT 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:

\[
D(x, g) = \begin{cases} 
\frac{1}{3\Sigma_{\text{trans}}(x, g)} & \text{If } \Sigma_{\text{trans}}(x, g) \text{ is available.} \\
\frac{1}{3\Sigma_t(x, g)} & \text{If not,} \\
\frac{1}{3 \left[ \Sigma_t(x, g) - \Sigma_{s1}(x, g \rightarrow g) \right]} & \text{Isotropic scatter} \\
\text{Anisotropic scatter} & \text{(1)}
\end{cases}
\]

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, \(\Sigma_{\text{trans}}(x, g)\) is the macroscopic transport cross section (transport is normally provided on the ISOTXS...
or GRUPXS files), and $\Sigma_{s1}(x, g \to g)$ is the $P_1$ anisotropic self-scatter cross section. It should be noted that $\Sigma_s(x, g)$ 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:

$$-\nabla \cdot D_g \nabla \phi^k_g + \sigma_{R,g}(r) \phi^k_g(r) + \sum_{g'=1}^{g-1} \sigma_{s,g' \to g} \phi^{k+1}_{g'}(r) +$$

where $g$ is the energy group number,

$k$ is the diffusion sub-outer iteration index,

$\sigma_{R,g} = \sigma_{t,g} - \sigma_{s,g' \to g}$ is the removal cross section,

$$\Phi^k_g(r) = \sum_{g'=1}^{G} (\psi f)_{g'} \phi^k_{g'}(r)$$

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:

$$\Omega \cdot \nabla \psi^{l+1/2}_g(r, \Omega) + \sigma_{t,g}(r) \psi^{l+1/2}_g(r, \Omega) = \sigma_{s,g' \to g} \phi^{l+1}_{g'}(r) +$$

where

$$g = 1, ..., G$$
where $l$ is the inner iteration index,

$$\psi_g(r, \Omega)$$

is the angular flux for group $g$,

$$\phi_g^l(r) = \int \psi_g^l(r, \Omega) d\Omega$$

is the transport scalar flux,

$$Q_g(r) = \frac{\chi_g \Phi^{CD}(r)}{k_{eff}} + \sum_{g' = g+1}^{G} \sigma_{s, g' \rightarrow g} \phi_g^{CD}(r)$$

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, $\phi_g^{l+1}$, comes from the solution of the DSA equation for this group, i.e.:

$$-\nabla \cdot D_g^{l+1/2} \nabla \phi_g^{l+1} + \sigma_{r,g} \phi_g^{l+1}(r) = \sum_{g' = 1}^{g-1} \sigma_{s, g' \rightarrow g} \phi_g^{l+1}(r) + Q_g(r) \quad (4)$$

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, $\phi_g^{l+1}(r)$. 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.

MORE DETAILS ON SOLVER INPUT

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:

\[
\max \left| \frac{\phi_{i,g}^l + 1 - \phi_{i,g}^l}{\phi_{i,g}^l} \right| < \text{EPSI} ,
\]

where \( \phi_{i,g}^l \) is the scalar flux for mesh point \( i \), group \( g \), and inner iteration \( l \), and \( \text{EPSI} \) 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

\[
\max_i \left| \frac{\Phi_{i}^{k+1} - \Phi_{i}^{k}}{\Phi_{i}^{k}} \right| < 3.0*\text{EPS} \quad (6)
\]

and

\[
\left| 1 - \lambda_D^{k+1} \right| < \text{EPS} .
\]

where

\( \Phi^k \) is the fission distribution at sub-outer \( k \),

\[ EPS = \max (\text{EPSI}, 0.01*(0.1**p) ) , \]
The notation \((F,G)\) denotes the inner product, or volume integral, of the product \(F\ast G\). In the above, \(\lambda_{D}^{k+1}\), 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 \(\text{EPSI}\).

**Transport Source Iteration Convergence.**

The convergence of the transport source triggers the increase of the number of transport inners from \(\text{IITL}=1\) to \(\text{IITL}=\text{IITM}\) 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

\[
\max_{i} \left| \frac{\Phi_{i}^{p*} - \Phi_{i}^{p}}{\Phi_{i}^{p}} \right| < 10.0*\text{EPSI} \tag{8}
\]

and

\[
|1 - \lambda_{p*}^{p*}| < 1.5*\text{EPSI} , \tag{9}
\]

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,

\[
\lambda_{p*}^{p*} \equiv \left( \frac{\Phi_{p*}^{p*}, 1}{\Phi_{p}^{p}, 1} \right) . \tag{10}
\]

**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

\[ |1 - \lambda^{p+1}| < \text{EPSI} , \]

and

\[ \max_i \left| \frac{\phi_i^{p+1} - \phi_i^p}{\phi_i^p} \right| < \text{EPSI} \] (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-
thetis 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,

\[ \psi_{\text{in}}(m) = \frac{\sum w_n \mu_n \psi(\mu_n)_{\text{out}}}{\sum w_n \mu_n} \]

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

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

Note: Use of a reflective boundary condition (IBL or IBR=1) requires the S_n quadrature set to be symmetric about \( \mu = 0 \).
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 IBL1/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, 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 Sn 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 $D_P_N$ 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 $D_P_N$ 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 Sn 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 $1 - \sum w_m \sum \mu_m$, and $\sum w_m \mu_m$
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
TWOTIL4N 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 possi-
bile to rotations about the coordinate axes. There is also available a Gauss-Chebychev set
triggered by the Block-V input variable IQUAD = 121. 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 $S_{n-1}$. 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, $\chi_g$, $g=1, 2 \ldots, NGROUP$. From certain of the
cross-section libraries (ISOTXS, GRUPXS, MENDF, etc.) a single $\chi$ vector characteriz-
ing the dominant fissionable isotope ($U_{235}$, $U_{238}$, $Pu_{239}$, etc.) can be extracted automa-
tically from the library and used for the problem. Alternatively, a single $\chi$ vector can
be provided by the user in Block-III through the card-image CHIVEC input array.
Again, this is a single $\chi$ 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 $\chi$ 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 $\chi$'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 $\chi$ vectors $(1 \leq N \leq NZONE)$ with each vector containing $NGROUP$ $\chi$ values.
The first $\chi$ 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 $\chi$ vector is applied to zone num-ber 2 (defined by the second entry in the ASSIGN array), etc. If the total number of $\chi$
 vectors in the CHI array is less than NZONE (the total number of zones), then the last $\chi$
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 $\chi$'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 FIXSRC3 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):

\[ Q_g(r, \Omega) = \sum_{n=1}^{NMQ} (2L+1) R_n(\Omega) \tilde{Q}_{n,g}(r) \quad , \quad g = 1, ..., NGROUP \]  

(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 \((g,n)\ast SOURCX(r,n) = \tilde{Q}_{n,g}(r)\). 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 \( \mu \) is replaced by \( \xi \). The six moments to be supplied in the input are thus:

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

(13a)

\[
\tilde{Q}_{2, g}(r) = \frac{1}{4\pi} \int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi P_{1}(\xi) \cos \phi Q_{g}(r, \xi, \phi) = Q_{c, 1, g}(r),
\]

(13b)

\[
\tilde{Q}_{3, g}(r) = \frac{1}{4\pi} \int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi P_{2}(\xi) Q_{g}(r, \xi, \phi) = Q_{2, g}(r),
\]

(13c)

\[
Q_{4, g}(r) = \frac{1}{4\pi} \int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi \sqrt{3} P_{2}(\xi) \cos 2\phi Q_{g}(r, \xi, \phi) = Q_{c, 2, g}(r),
\]

(13d)

\[
\tilde{Q}_{5, g}(r) = \frac{1}{4\pi} \int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi \sqrt{6} P_{3}(\xi) \cos \phi Q_{g}(r, \xi, \phi) = Q_{c, 3, g}(r),
\]

(13e)

and

\[
\tilde{Q}_{6, g}(r) = \frac{1}{4\pi} \int_{-1}^{1} d\xi \int_{0}^{2\pi} d\phi \sqrt{10} P_{3}(\xi) \cos 3\phi Q_{g}(r, \xi, \phi) = Q_{c, 3, g}(r),
\]

(13f)

for \( n=1, \ldots, \text{NGROUP} \).

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

It is worth noting that most inhomogeneous distributed sources are assumed to be isotropic, so that \( \text{NMQ} \) in Eq. (12) is unity and the only source moment entered is the zeroth moment.

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

(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, $Q_m$:

$$\psi_{\text{incoming}} (\mu_m) = Q_m.$$

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 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, SAFRTN, SABACK). In ONEDANT, the order of SARITE and SALEFT entries is as follows:

$$\psi_{1,1}, \psi_{2,1}, \psi_{3,1}, \ldots; \psi_{1,2}, \psi_{2,2}, \psi_{3,2}, \ldots; \ldots$$

- Slab Geometry (IGEOM= 1 or SLAB or PLANE):

The ordering of the angles is as shown below (S6 used for illustration).

![Figure 7.2 Ordering in slab geometry.](image-url)
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.

<table>
<thead>
<tr>
<th>$m$</th>
<th>SARITE $\mu$</th>
<th>SALEFT $\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mu_1$</td>
<td>$\mu_4$</td>
</tr>
<tr>
<td>2</td>
<td>$\mu_2$</td>
<td>$\mu_5$</td>
</tr>
<tr>
<td>3</td>
<td>$\mu_3$</td>
<td>$\mu_6$</td>
</tr>
</tbody>
</table>

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 $\mu_1$ while the first entry in SALEFT (for each group) corresponds to $\mu_4$ and $|\mu_1| \neq |\mu_4|$. 
• Cylindrical Geometry (IGEOM= 2 or CYLINDER or CYL):

Using S₆ for illustrative purposes, the principal octant (μ > 0, η > 0) of the unit sphere of angular directions is as shown below.

![Figure 7.3 Quadrature points in cylindrical geometry.](image)

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>
<thead>
<tr>
<th>m</th>
<th>μ</th>
<th>η</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-μ₁</td>
<td>η₁</td>
</tr>
<tr>
<td>2</td>
<td>-μ₂</td>
<td>η₂</td>
</tr>
<tr>
<td>3</td>
<td>-μ₁</td>
<td>η₂</td>
</tr>
<tr>
<td>4</td>
<td>-μ₃</td>
<td>η₃</td>
</tr>
<tr>
<td>5</td>
<td>-μ₂</td>
<td>η₃</td>
</tr>
<tr>
<td>6</td>
<td>-μ₁</td>
<td>η₃</td>
</tr>
</tbody>
</table>

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 ≥ 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 ≤ ISCT ≤ 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

\[ \hat{\sigma}_s^l (g \rightarrow g) = \sigma_s^l (g \rightarrow g) - \sum_{g'} \sigma_s^{ISCAT + 1} (g \rightarrow g'), \; l = 0, 1, \ldots, ISCT. \]

• 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 $\sigma'_{s}(g \rightarrow g')$ is unity. Generally, the Cesàro transport correction should only be used with ISCT $\geq 2$.

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

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

**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 $\kappa_{eff}$. The type of search is controlled by the input parameter IEVT supplied in Block-V of the card-image input as follows:
IEVT\textsuperscript{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\textsubscript{eff} calculations.

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

\[ \psi (r, \Omega, t) = \psi (r, \Omega) e^{\alpha t} \]

If this assumption is inserted into the time-dependent transport equation, the exponentials cancel and a fictitious cross-section term of the form \( \alpha / v_g \) 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 \( \alpha \) 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, \( R_k \)

\[ \tilde{R}_{k+1} = \tilde{R}_k + (R_{k+1} - R_k) \times (1 + EV \times RM_k) \quad k=1,2,\ldots,IM \quad (15) \]

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 $k_{\text{eff}}$. 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 $k_{\text{eff}}$ 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 $\alpha$ as the parametric value PV in the input. Note that an $\alpha$ of 0.0 corresponds to a normal concentration or dimension search on a $k_{\text{eff}}$ 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 $k_{\text{eff}}$ 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 $\lambda$ unity where $\lambda$ is defined as

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

for the $k^{\text{th}}$ 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 $\lambda$ from unity for each outer iteration is plotted.

For the initial system, NEWPAR continues the outer iteration until two successive values of $\lambda$ differ by less than EPSO. For subsequent sequences of $\lambda$ values, a different convergence precision, XLAX, is used. After the first converged $\lambda$ 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 $\lambda$ 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 $\lambda$ 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
MORE DETAILS ON SOLVER INPUT

Eigenvalue Searches

limit (XLAH). If a parabolic fit is tried and the roots are imaginary, a straight-line fit is used. If the roots are not imaginary, the closest root is used as the new value of EV. Once a bracket is obtained (change of sign of \( \lambda - 1 \)), the fit procedure is not allowed to move outside the region of the bracket. Should a parabolic fit select an eigenvalue outside the bracket region, this value is rejected and the new value is taken to be one-half the sum of the previous value and the value previous to that.

Whenever the parabolic fit is not used, a linear fit is used and the new eigenvalue is computed from

\[
(EV)_{\text{new}} = (EV)_{\text{old}} + \text{POD} \times \text{EVS} \times (1 - \lambda),
\]

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 \( |1 - \lambda| < XLAH \), the value of EVS is fixed and kept constant until convergence to prevent numerical difficulty in the approximation of the derivative when \( \lambda \) is close to unity.

Because parametric search problems represent sequences of \( k_{\text{eff}} \) 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 $\lambda$ during a hypothetical eigenvalue search.

Convergence in time-absorption calculations is typically one-sided. If EV (the eigenvalue $\alpha$) 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 $\chi_x \Sigma (\nu \sigma_f) \phi_h$, one has $(\nu \sigma_f) \sum \chi_h \phi_h$. 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 $\Omega$ must be identified as the adjoint solution for direction $\Omega$. For example, the vacuum boundary condition at a surface (no incoming angular flux) in an adjoint calculation must be interpreted as a condition of no outgoing flux. Likewise, the adjoint leakage at a surface must be interpreted as incoming instead of outgoing.

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

The printed output from the solver module in an adjoint calculation indicates the correct group ordering and need not be inverted by the user. The adjoint fluxes from the solver module are written to a binary ATFLUX\textsuperscript{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


RUNNING THE EDIT MODULE

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory
# TABLE OF CONTENTS

**TABLE OF CONTENTS** .............................................................................................. 8-3

**LIST OF TABLES** ........................................................................................................ 8-5

**INTRODUCTION** ........................................................................................................ 8-7

**REACTION RATES** ..................................................................................................... 8-9
  - Spatial Options for Edits .................................................................................................. 8-9
  - Energy Group Options For Edits ...................................................................................... 8-10
  - Forms Of Response Functions ........................................................................................ 8-11
    - Cross-Section Response Functions: EDXS ................................................................. 8-11
    - User-Input Response Functions: RSFE, RSFX ............................................................ 8-13
  - Response Function Summing Options .............................................................................. 8-13
    - Cross-Section Response Functions Sums: MICSUM .................................................. 8-13
    - User-Input Response Functions Sums: IRSUMS ......................................................... 8-14
  - Adjoint Edits .................................................................................................................. 8-15
  - ASCII File Output Capabilities (the EDOUTF Parameter) ............................................... 8-15

**MASS INVENTORIES** ................................................................................................... 8-17
LIST OF TABLES

Table 8.1: MASSED Input Values ......................................................................... 8-17
Table 8.2: MASSED Input Values for Fine Mesh Mix Problem ........................... 8-18
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 \( \rho \), for the \( i \)th spatial mesh point is computed as

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

where:

- \( \phi_{i,g} \) = scalar flux for mesh point \( i \), energy group \( g \).
- \( R_{i,g} \) = a response function which may be either input directly via the RSFE and RSFX arrays (below) or formed from input cross sections.
- \( g' \) denotes an Edit energy-broad-group (See "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, TECPLPOT® which is commercially available, the edit module also has the capability to generate TECPLPOT files which mirror the printed output above. This capability is available 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 / TECPLPOT files only / both generate the TECPLPOT 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, $\rho$, for Edit Zone $Z_m$ as

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

If Edit Zone edits are requested (ZNED=1) and the EDZONE array is not specified, the code will assume a default specification of the Edit Zones equal to the Coarse-Mesh intervals (see 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, $(n,\gamma)$, $(n,\alpha)$, 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 $n,\alpha$ and $n,\gamma$ 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, $\Sigma^{RES}_{i,g}$ 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

   $$ R_{i,g} = \Sigma^{RES}_{i,g} $$

   the Block-VI input parameter RESDNT is set to unity.

2. Isotope Microscopic Cross-Section Response Functions: EDISOS Input Array.

   Isotope microscopic cross section, $\sigma^{ISO}_{g}$, 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

   $$ R_{i,g} = \sigma^{ISO}_{g} $$

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


   Resident constituent cross sections, $\Sigma^{ISO}_{i,g}$, 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

   $$ R_{i,g} = N^{ISO}_{i} \sigma^{ISO}_{g} = \Sigma^{ISO}_{i,g} $$

   for spatial mesh interval $i$, group $g$.


   Material macroscopic cross sections, $\Sigma^{MATL}_{g}$, 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

\[ R_{i,g} = \sum_{g}^{MATL} \]

Note that these response functions are spatially constant so that the edit quantity \( \sum_{g}^{MATL} \phi_{i,g} \) 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,

\[ R_{i,g} = RSFE_g \times RSFX_i \]

**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

\[
\text{EDCONS} = \text{PU239, PU240, PU241, U238,}
\]

and the MICSUM array were specified as

\[
\text{MICSUM} = \text{PU239, PU241, 0, "N-GAMM," "N-FISS," 0, PU240, U238, 0, ABS}
\]

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

\[
\begin{align*}
\text{SUM 1:} & \left\{ \left[ N_i (\sigma_n^g + \sigma^f_g) \right]_{\text{PU239}} + \left[ N_i (\sigma_n^g + \sigma^f_g) \right]_{\text{PU241}} \right\} \times \phi_{i,g} \\
\text{SUM 2:} & \left[ (N_i \sigma_a^g)_{\text{PU240}} + (N_i \sigma_a^g)_{\text{U238}} \right] \times \phi_{i,g}.
\end{align*}
\]

**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.
REACTION RATES

ASCII File Output Capabilities (the EOUTF Parameter)
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

<table>
<thead>
<tr>
<th>MASSED</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>1a</td>
<td>Edit by Solver zones</td>
</tr>
<tr>
<td>2</td>
<td>Edit by Edit zonesb</td>
</tr>
<tr>
<td>3</td>
<td>Edit by both Solver and Edit zones</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>MASSED</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>1&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Total inventory for the whole problem</td>
</tr>
<tr>
<td>2</td>
<td>Inventories by Edit zones&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>3</td>
<td>Inventories for both Total and Edit zones</td>
</tr>
</tbody>
</table>

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.
TABLE OF CONTENTS

TABLE OF CONTENTS ........................................................................................................ 9-3
LIST OF TABLES ............................................................................................................... 9-5
INTRODUCTION ................................................................................................................. 9-7
FREE FIELD DEFINITIONS .............................................................................................. 9-9
  Arrays .............................................................................................................................. 9-9
  Numeric Data Items ..................................................................................................... 9-9
  Character Data Items ................................................................................................... 9-10
  Blocks ............................................................................................................................ 9-10
  Strings ......................................................................................................................... 9-10
  Comments .................................................................................................................... 9-10
  Operators ..................................................................................................................... 9-10
FREE FIELD INPUT DETAILS ......................................................................................... 9-13
  Free-Field Input ........................................................................................................... 9-13
  User-Specified Input Formats ...................................................................................... 9-16
    U Operator ................................................................................................................ 9-16
    V Operator ................................................................................................................ 9-17
FIXED-FIELD FIDO INPUT DETAILS ........................................................................... 9-19
  Fixed-Field FIDO Input ................................................................................................ 9-19
REFERENCES ................................................................................................................ 9-23
# LIST OF TABLES

Table 9.1: Free Field Data Operators ..................................................................... 9-15
Table 9.2: Special Fixed Field Data Operators ...................................................... 9-21
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.
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:

\[ \text{SOURCX} = 0 \; 0 \; 0 \; 1.1 \; 1.1 \; 0 \; 0 \; 0 \; 0 \; 0 \]

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

<table>
<thead>
<tr>
<th>Data Numerator</th>
<th>Data Operator</th>
<th>Data Entry</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>R</td>
<td>d</td>
<td>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</td>
</tr>
<tr>
<td>n</td>
<td>I</td>
<td>d</td>
<td>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.</td>
</tr>
<tr>
<td>n</td>
<td>L</td>
<td>d</td>
<td>LOGARITHMIC INTERPOLATE OPTION: The effect is the same as that of &quot;I&quot; except that the resulting interpolates are equally separated in log-space.</td>
</tr>
<tr>
<td>-</td>
<td>F</td>
<td>d</td>
<td>FILL OPTION: Fill the remainder of the data string with the value d.</td>
</tr>
<tr>
<td>n</td>
<td>Z</td>
<td>-</td>
<td>ZERO OPTION: Enter the value zero in the data string n successive times. Example: 4Z → 0 0 0 0</td>
</tr>
<tr>
<td>n</td>
<td>S</td>
<td>-</td>
<td>SKIP OPTION: Causes the pointer to skip n positions in the current string leaving the data values in those positions unchanged.</td>
</tr>
<tr>
<td>n</td>
<td>Q</td>
<td>m</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Table 9.1 Free Field Data Operators (Cont.)

<table>
<thead>
<tr>
<th>Data Numerator</th>
<th>Data Operator</th>
<th>Data Entry</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>n G m</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n N m</td>
<td>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 5 4 4 5 6 can be input as 4 5 6 2N3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n M m</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n Y m</td>
<td>STRING REPEAT OPTION: Enter the preceding m strings of data into the current array n successive times. (For multistring arrays only.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n X -</td>
<td>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.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>n C d</td>
<td>SEQUENCE MULTIPLY OPTION: Enter the last n data entries, multiplied by d, into the current data string.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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.
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, \( \leq 99 \)) 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>
<thead>
<tr>
<th>Data Numerator</th>
<th>Data Operator</th>
<th>Data Entry</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>*</td>
<td></td>
<td>FLOATING ARRAY IDENTIFIER. The unique array number is n. Precedes the data in the array.</td>
</tr>
<tr>
<td>n</td>
<td>+ d</td>
<td></td>
<td>POSITIVE EXPONENTIATION. This will enter a single number with magnitude $d \times 10^n$, 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.</td>
</tr>
<tr>
<td>n</td>
<td>- d</td>
<td></td>
<td>NEGATIVE EXPONENTIATION. This will enter a single number with magnitude $d \times 10^{-n}$, 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.</td>
</tr>
</tbody>
</table>
CROSS-SECTION LIBRARIES

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory
# TABLE OF CONTENTS

**TABLE OF CONTENTS** ............................................................................................ 10-3

**LIST OF TABLES** ...................................................................................................... 10-5

**INTRODUCTION** ...................................................................................................... 10-7

**INPUT OF THE BASIC CROSS-SECTION LIBRARY** ........................................... 10-9
  ISOTXS and GRUPXS Standard Interface Files ......................................................... 10-9
  Card-Image Libraries in Los Alamos, ANISN, or Free Field Format ......................... 10-9
    Ordering of Cross Sections Within a Cross-Section Table ...................................... 10-10
    Card-Image Data Formats .......................................................................................... 10-11
    Cross-Section Table Title Cards ........................................................................... 10-11
    Anisotropic Scattering and the Ordering of Cross-Section Tables ......................... 10-11
  Binary Form of Card-Image Libraries (the BXSLIB file) ........................................... 10-12
  XSLIBB Card-Image Library File ............................................................................... 10-12
  MACRXS and SNXEDT Cross-Section Files ............................................................... 10-13
  The MACBCD Card-Image Cross-Section Library .................................................. 10-13
  The Los Alamos MENDF5 Cross-Section Library .................................................. 10-13
  The Los Alamos MENDF5G Gamma Cross-Section Library .................................. 10-14
  The XSLIBE and XSLIBF Material Cross-Section Libraries .................................. 10-14

**COUPLED NEUTRON-GAMMA CROSS SECTIONS** ......................................... 10-15

**CREATING FILES WITH DIFFERENT FORMATS** ............................................. 10-19

**BALANCING THE CROSS SECTIONS** ................................................................. 10-21

**REFERENCES** ....................................................................................................... 10-23
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
</tr>
</thead>
</table>

10-4 CROSS-SECTION LIBRARIES Version 3.0
LIST OF TABLES

Table 10.1: Cross-Section Ordering In A Card-Image Library ......................... 10-10
Table 10.2: Arrangement of Data in a Coupled Neutron-Gamma Library Table ... 10-16
Table 10.3: Coupled Cross-Section Table Example ....................................... 10-17
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. 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 $\sigma_t$ (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

<table>
<thead>
<tr>
<th>Position in the Row</th>
<th>Cross-Section Type For Group g</th>
<th>Collective Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma_a$</td>
<td>Edit Positions, Optional</td>
</tr>
<tr>
<td>IHT-2</td>
<td>$\nu \sigma_f$</td>
<td>Principal Cross Sections</td>
</tr>
<tr>
<td>IHT*</td>
<td>$\sigma_t$</td>
<td>Upscatter Cross Sections</td>
</tr>
<tr>
<td></td>
<td>$\sigma_s(g+2\rightarrow g)$</td>
<td></td>
</tr>
<tr>
<td>IHS-2</td>
<td>$\sigma_s(g+1\rightarrow g)$</td>
<td></td>
</tr>
<tr>
<td>IHS*</td>
<td>$\sigma_s(g\rightarrow g)$</td>
<td>Self</td>
</tr>
<tr>
<td></td>
<td>$\sigma_s(g-1\rightarrow g)$</td>
<td>Downscatter Cross Sections</td>
</tr>
<tr>
<td></td>
<td>$\sigma_s(g-2\rightarrow g)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sigma_s(g-3\rightarrow g)$</td>
<td></td>
</tr>
<tr>
<td>IHM*</td>
<td></td>
<td>Last Position</td>
</tr>
</tbody>
</table>

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

\[
\sigma_s (g' \rightarrow g, \mu_0) = \sum_{n=0}^{ISCT} \frac{2n + 1}{4\pi} P_n (\mu_0) \sigma_s^n (g' \rightarrow g),
\]

where \( \mu_0 = \Omega' \cdot \Omega \), 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 ≤ 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 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-I 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.
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 photon-neutron, 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

GROUP
ROW  1 2 . LNG LNG+1 . NGROUP

<table>
<thead>
<tr>
<th>1 2 . IHT IHS : IHS+LNG-1 IHS+LNG . . IHM=IHT+NGROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINCIPAL CROSS SECTIONS AND EDIT POSITIONS</td>
</tr>
<tr>
<td>NEUTRON PRODUCTION VIA NEUTRON SCATTER</td>
</tr>
<tr>
<td>NOT USED</td>
</tr>
<tr>
<td>GAMMA PRODUCTION VIA GAMMA SCATTER</td>
</tr>
<tr>
<td>GAMMA PRODUCTION VIA NEUTRON SCATTER</td>
</tr>
</tbody>
</table>
### Table 10.3  Coupled\(^a\) Cross-Section Table\(^b\) Example

<table>
<thead>
<tr>
<th>NEUTRON GROUPS</th>
<th>GAMMA GROUPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (\sigma_n(1))</td>
<td>(\sigma_n(5))</td>
</tr>
<tr>
<td>2 (\nu\sigma_n(1))</td>
<td>0</td>
</tr>
<tr>
<td>3 (\sigma_c(1))</td>
<td>(\sigma_c(5))</td>
</tr>
<tr>
<td>4 (\sigma(1\rightarrow1))</td>
<td>(\sigma(5\rightarrow5))</td>
</tr>
<tr>
<td>5 0</td>
<td>(\sigma(5\rightarrow6))</td>
</tr>
<tr>
<td>6 0</td>
<td>(\sigma(5\rightarrow7))</td>
</tr>
<tr>
<td>7 0</td>
<td>(\sigma(4\rightarrow7))</td>
</tr>
<tr>
<td>8 0</td>
<td>(\sigma(3\rightarrow7))</td>
</tr>
<tr>
<td>9 0</td>
<td>(\sigma(2\rightarrow7))</td>
</tr>
<tr>
<td>10 0</td>
<td>(\sigma(1\rightarrow7))</td>
</tr>
</tbody>
</table>

\(^a\) 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:

\[
\sigma_{r,g} = \sigma_{\text{total},g} - \sigma_{g\rightarrow g}
\]

(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 \( \sigma_{\text{total},g} \) and \( \sigma_{g\rightarrow g} \) (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)

\[
\sigma_{out,g} = \sigma_{r,g} - \sigma_{\text{abs},g}
\]

(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:

\[ \sigma_{abs,g} + \sum_{g'} \sigma_{g \rightarrow g'} = \sigma_{total,g} \]  

(22)

Again, recall that the transfer cross sections, \( \sigma_{g \rightarrow g'} \), 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:

\[ \sigma'_{abs,g} = \sigma_{total,g} - \sum_{g'} \sigma_{g \rightarrow g'} \]  

(23)

Where we are now using the only transfer cross sections we have, i.e., the inscattering cross sections, for \( \sigma_{g \rightarrow g'} \). 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


MATERIAL MIXING TUTORIAL

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>11-3</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>11-5</td>
</tr>
<tr>
<td>REVIEW OF TERMINOLOGY</td>
<td>11-7</td>
</tr>
<tr>
<td>MIXING MATERIALS FROM &quot;ISOTOPES&quot;</td>
<td>11-9</td>
</tr>
<tr>
<td>ASSIGNING MATERIALS TO ZONES</td>
<td>11-11</td>
</tr>
<tr>
<td>ALTERNATIVE FORMS OF MIXING</td>
<td>11-13</td>
</tr>
<tr>
<td>Using Atomic Fractions or Weight Fractions (MATSPEC)</td>
<td>11-13</td>
</tr>
<tr>
<td>Providing Atomic Weights to the Code (the ATWT Array)</td>
<td>11-15</td>
</tr>
<tr>
<td>INTERFACE FILES USED IN MIXING</td>
<td>11-17</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>11-19</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 11.1: Example Specification of Materials........................................................................ 11-10
Table 11.2: Composition of Zones......................................................................................... 11-11
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.
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. 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

<table>
<thead>
<tr>
<th>Material Identifier</th>
<th>Isotope Identifier</th>
<th>Atom Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS</td>
<td>CHROME</td>
<td>1.6676E-2</td>
</tr>
<tr>
<td></td>
<td>NICKEL</td>
<td>6.566E-3</td>
</tr>
<tr>
<td></td>
<td>IRON</td>
<td>6.3832E-2</td>
</tr>
<tr>
<td>UO2</td>
<td>U235</td>
<td>6.7E-5</td>
</tr>
<tr>
<td></td>
<td>U238</td>
<td>2.223E-2</td>
</tr>
<tr>
<td></td>
<td>OXYGEN</td>
<td>4.460E-2</td>
</tr>
<tr>
<td>PUO2</td>
<td>PU239</td>
<td>1.7761E-2</td>
</tr>
<tr>
<td></td>
<td>PU240</td>
<td>4.440E-3</td>
</tr>
<tr>
<td></td>
<td>OXYGEN</td>
<td>4.4402E-2</td>
</tr>
<tr>
<td>NA</td>
<td>SODIUM</td>
<td>2.096E-2</td>
</tr>
</tbody>
</table>

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 2.223-2, OXYGEN 4.460-2;
    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>
<thead>
<tr>
<th>Zone</th>
<th>Material</th>
<th>Material Volume Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE</td>
<td>SS</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>NA</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>U8O2</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>PU9O2</td>
<td>0.15</td>
</tr>
<tr>
<td>BLKT</td>
<td>SS</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>NA</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>U8O2</td>
<td>0.349</td>
</tr>
<tr>
<td></td>
<td>U5O2</td>
<td>0.001</td>
</tr>
<tr>
<td>REFL</td>
<td>SS</td>
<td>0.030</td>
</tr>
<tr>
<td></td>
<td>NA</td>
<td>0.70</td>
</tr>
</tbody>
</table>

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.
ASSIGNING MATERIALS TO ZONES

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.
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-D of the DANTSYS input (the mixing specifications) is the parameter MATSPEC. There are three allowable input values for MATSPEC:

a. MATSPEC=A~ENS 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=WTFRA C 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.

$\rho_m$ 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

  \[
  \text{MATLS= mat}_1 \ iso_{1,1} \ af_{1,1} \ iso_{2,1} \ af_{2,1} \ldots; \ mat_2 \ iso_{1,2} \ af_{1,2} \ iso_{2,2} \ af_{2,2} \ldots; \ldots; \text{etc.}
  \]

  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

  \[
  \text{ASSIGN= zonid}_1 \ mat_i \ \rho_i \ldots; \ zonid_2 \ mat_k \ \rho_k \ mat_l \ \rho_l \ldots; \text{etc.}
  \]

  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

\[
\text{MATLS=} \text{mat}_1 \text{ iso}_{1,1} \text{ wf}_{1,1} \text{ iso}_{2,1} \text{ wf}_{2,1} \ldots; \text{mat}_2 \text{ iso}_{1,2} \text{ wf}_{1,2} \text{ iso}_{2,2} \text{ wf}_{2,2} \ldots; \text{etc.}
\]

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

\[
\text{ASSIGN=} \text{zonid}_1 \text{ mat}_1 \rho_1 \ldots; \text{zonid}_2 \text{ mat}_k \rho_k \text{ mat}_l \rho_l \ldots; \text{etc.}
\]

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:

\[
\text{MATSPEC=} \text{WTFRAC ATFRAC}
\]

\[
\text{MATLS=} \text{SS, IRON 0.74, CHROME 0.18, NICKEL 0.08; UO2, U235 0.003, U238 0.997, OXYGEN 2.0; PUO2, PU239.8, PU240 0.2, OXYGEN 2.0; NA, SODIUM 1.0;}
\]

\[
\text{ASSIGN=} \text{STEEL SS 8.0; UFUEL UO2 210.0; PFUEL PUO2 210.0; COOLNA 0.0;}
\]

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
ALTERNATIVE FORMS OF MIXING

Providing Atomic Weights to the Code (the ATWT Array)

\[ \text{ATWT} = iso_1 \ atwt_1 \ iso_2 \ atwt_2 \ \ldots \ iso_n \ atwt_n \]

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.
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


ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT – Methods Manual

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory
# TABLE OF CONTENTS

## TABLE OF CONTENTS

- TABLE OF CONTENTS .................................................. 12-3
- LIST OF FIGURES ...................................................... 12-5
- LIST OF TABLES .......................................................... 12-7
- INTRODUCTION .......................................................... 12-9
  - The First Order Form of the Boltzmann Transport Equation ........................................ 12-9
- MULTIGROUP, DISCRETE ORDINATES TRANSPORT THEORY .................................. 12-11
  - Multigroup Approximation ........................................... 12-11
  - Discrete Ordinates Approximation ................................ 12-12
  - Iteration Procedure and Diffusion Synthetic Acceleration ........................................... 12-14
- ONEDANT METHODS ................................................. 12-19
  - Geometrical Symmetries Treated in ONEDANT ......................................................... 12-19
  - Particular Forms of the Divergence Operator .............................................................. 12-21
  - Spherical Harmonics Expansion of the Scattering Source .......................................... 12-22
  - Spherical Harmonics Expansion of the Inhomogeneous Source .................................. 12-26
  - Discrete-Ordinates Equations in One Dimension ....................................................... 12-27
  - Discretization of the Spatial Variable .......................................................................... 12-32
- TWODANT METHODS .................................................. 12-35
  - Some Angular Details in TWODANT .............................................. 12-35
  - Transport Operator in Two-Dimensional Symmetries ................................................. 12-36
    - Planar X-Y Symmetry .............................................................................................. 12-36
    - Planar R-Φ Symmetry ............................................................................................. 12-37
    - Cylindrical R-Z Symmetry ...................................................................................... 12-37
  - Spatially Discretized Two-Dimensional Transport Equation ........................................ 12-37
  - Monte Carlo/Discrete Ordinates Hybrid Method ....................................................... 12-40
    - Overview .................................................................................................................. 12-40
    - TWODANT/MC Input ............................................................................................... 12-40
    - The Monte Carlo region ........................................................................................... 12-41
    - Source Definition ..................................................................................................... 12-41
    - Number of Histories and Code Flow ........................................................................ 12-42
    - Convergence Criteria .............................................................................................. 12-43
    - Memory Requirements ............................................................................................ 12-45
    - Treatment of Angular Scattering ............................................................................. 12-45
    - Variance Reduction ................................................................................................. 12-45
    - Statistics .................................................................................................................. 12-47
    - Edits ......................................................................................................................... 12-47
- TWODANT/GQ METHODS ............................................ 12-49
- TWOHEX METHODS .................................................... 12-51
- THREEDANT METHODS ................................................ 12-53
- REFERENCES ............................................................... 12-55
LIST OF FIGURES

Figure 12.1: Coordinates in plane geometry ........................................................... 12-19
Figure 12.2: Coordinates in cylindrical geometry .................................................. 12-20
Figure 12.3: Coordinates in spherical geometry ..................................................... 12-20
Figure 12.4: Ordering of $S_6$ directions in plane and spherical geometries .......... 12-28
Figure 12.5: Ordering of $S_4$ directions in two-angle plane geometry ................. 12-29
Figure 12.6: Ordering of $S_6$ directions in cylindrical geometry .......................... 12-30
LIST OF TABLES

Table 12.1: Forms of the Divergence Operator ............................................. 12-21
Table 12.2: Number of Spherical Harmonics, N, as a Function of Order .......... 12-24
Table 12.3: Spherical Harmonics, for Different Geometries ...................... 12-25
Table 12.4: Number of Quadrature Points, M as a Function of S_n Order, N .... 12-27
Table 12.5: Spherical Harmonics in Two Dimensions ................................... 12-35
Table 12.6: Number of Angles Per Octant in Two Dimensions .................. 12-36
INTRODUCTION

This chapter provides the development of the multigroup, discrete-ordinates, form of the time-independent Boltzmann transport equation. 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. 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:

\[ \nabla \cdot \vec{\Omega} \psi (r, E, \Omega) + \sigma (r, E) \psi (r, E, \Omega) = \int dE' d\Omega' \sigma_s (r, E' \rightarrow E) \psi (r, E', \Omega') \\
+ \frac{1}{k_{\text{eff}}} \int dE' d\Omega' \chi (r, E' \rightarrow E) \nabla \psi (r, E', \Omega') \\
+ Q (r, E, \Omega), \tag{1} \]

where \( \psi (r, E, \Omega) \) is the particle flux (particle number density times the particle speed defined such that \( \psi (r, E, \Omega) dE \, dr \, d\Omega \) is the flux of particles in the energy range \( dE \) about \( E \), in the volume element \( dr \) about \( r \), with directions of motion in the solid angle element \( d\Omega \) about \( \Omega \). Similarly, \( Q (r, E, \Omega) dE \, dr \, d\Omega \) is the rate at which particles are produced in the same element of phase space from sources that are independent of the flux \( \psi \). The macroscopic total cross section is \( \sigma \). Assuming isotropic media, the macroscopic scattering transfer cross section, from energy \( E' \) to energy \( E \) through a scattering angle \( \Omega \cdot \Omega' \), is \( \sigma_s \). And the macroscopic fission cross section is \( \sigma_f \). All of the quantities may be spatially dependent. The number of particles emitted isotropically \( \left( \frac{1}{4\pi} \right) \) per fission is \( v \), and the fraction of these particles appearing in energy \( dE \) about \( E \) from fissions in \( dE' \) about \( E' \) is \( \chi (r, E' \rightarrow E) \). If \( Q \) is not zero in Eq. (1), then \( k_{\text{eff}} \) is set to 1; if \( Q \) is zero, then the problem is an eigenvalue problem and \( 1/k_{\text{eff}} \) 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.
INTRODUCTION

The First Order Form of the Boltzmann Transport Equation
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:

\[
\mathbf{Q} \cdot \nabla \psi_g(r, \Omega) + \sigma_{t,g}(r) \psi_g(r, \Omega) = \frac{\chi_g(r)}{k_{\text{eff}}} \Phi(r) + Q_g(r, \Omega) +
\]

\[
\sum_{g'=1}^{G} \int_{\Omega'} \sigma_{s,g' \rightarrow g}(r, \mu_0) \psi_{g'}(r, \Omega') d\Omega'
\]

\( g = 1, \ldots, G \)

where, \( \mu_0 = \Omega \cdot \Omega' \),

\[
\psi_g(r, \Omega) = \int_{\Delta E_g} \psi(r, \Omega, E) \, dE,
\]

\[
\sigma_{t,g}(r) = \left( \int_{\Delta E_g} \sigma_t(r, E) \psi(r, \Omega, E) \, dE \right) / \psi_g(r, \Omega),
\]

\[
\sigma_{s,g' \rightarrow g}(r, \mu_0) = \left( \int_{\Delta E_g' \Delta E_g} \sigma(r, E \rightarrow E', \mu_0) \psi(r, \Omega, E') \, dE \, dE' \right) / \psi_g(r, \Omega),
\]
\[ Q_g(r, \Omega) = \int_{\Delta E_z} Q(r, \Omega, E) dE, \]

\[ \chi_g(r) = \int_{\Delta E_z} \gamma(r, E) dE, \]

\[ \Phi(r) = \sum_{g=1}^{G} \left( \int_{\Delta E_z} \nu \sigma_f(r, E) \phi(r, E) dE \right) \phi_g(r), \]

\[ \phi_g(r) = \int \psi_g(r, \Omega) d\Omega. \]

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, \( \Omega_m \), 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 from 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:
\[
\sum_{m=1}^{M} w_m \Omega_m = 0 \text{ for symmetry, and}
\]
\[
\sum_{m=1}^{M} w_m \mu_m = \sum_{m=1}^{M} w_m \eta_m = \sum_{m=1}^{M} w_m \xi_m = \frac{1}{3} \text{ for the diffusion limit.}
\]

where the components of \( \Omega_m \) are: \( \Omega_m = \mu_m i + \eta_m j + \xi_m k \).

To obtain the discrete ordinates balance equation, we integrate Eq (2) over \( w_m \) (in this case we assume isotropic scattering for simplicity):

\[
[\Omega \cdot \nabla \psi_g (\tau, \Omega)]_m + \sigma_t g (\tau) \psi_{g,m} (\tau) = \frac{\chi_g (\tau)}{k_{\text{eff}}} \Phi (\tau) + Q_{g,m} (\tau) + \\
\sum_{g'=1}^{G} \sigma_{s0,g' \rightarrow g} (\tau) \phi_{g'} (\tau)
\]

where,

\[
\psi_{g,m} (\tau) = \int_{w_m} \psi (\tau, \Omega) d\Omega,
\]

\[
\phi_g (\tau) = \sum_{m=1}^{M} w_m \psi_{g,m} (\tau),
\]

\[
Q_{g,m} (\tau) = \sum_{l} \sum_{q=0}^{q} (2l+1) Y_l^q (\Omega_m) Q_{g,l}^q (\tau),
\]

\[
Q_{g,l}^q (\tau) = \int Y_l^q (\Omega) Q_g (\tau, \Omega) d\Omega,
\]

and the \( Y_l^q (\Omega) \) 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

\[ \Omega \cdot \nabla \tilde{\psi}^l_g (r, \Omega) + \sigma_g (r) \tilde{\psi}^l_g (r, \Omega) = \sigma_{s,g \rightarrow g} (r) \phi^{l-1}_g (r) + QQ_g (r). \]  \hspace{1cm} (4)

In Eq. (4), \( \tilde{\psi}^l_g (r, \Omega) \) is the angular flux for group g at the lth inner iteration using a scalar flux \( \phi^{l-1}_g (r) \) 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 \( \phi^l_g \) 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

\[ -\nabla \cdot D_g (r) \nabla \phi^l_g (r) + \sigma_{R,g} (r) \phi^l_g (r) = QQ_g (r) - R^l_g (r), \]  \hspace{1cm} (5)

where

\[ D_g (r) = \frac{1}{3\sigma_{tr,g}} (r), \]  \hspace{1cm} \[ \sigma_{R,g} (r) = \sigma_g (r) - \sigma_{s,g \rightarrow g} (r) \]

and the correction term is

\[ R^l_g (r) = \nabla \cdot J^l_g (r) + \nabla \cdot D_g (r) \nabla \phi^l_g (r). \]  \hspace{1cm} (6)

In Eq. (6),
\[ \tilde{\phi}^l_g (r) = \int d\Omega \tilde{\psi}^l_g (r, \Omega), \quad \tilde{J}^l_g (r) = \int d\Omega \Omega \tilde{\psi}^l_g (r, \Omega). \] (7)

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

The source correction scheme for the inner iteration proceeds as follows: using \( \phi_g^{l-1} \), 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 \( \phi_g^l \) 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

\[ \nabla \cdot \tilde{J}_g (r) + \sigma_{R_g} (r) \tilde{\phi}_g (r) = \mathcal{Q}_g, \]

which is the converged transport balance equation obtained also by integrating Eq. (4) over all \( \Omega \).

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:

\[
-\nabla \cdot D_g (r) \nabla \phi_g^{k+1} (r) + \sigma_{R_g} (r) \phi_g^{k+1} (r) = \mathcal{Q}_g (r) - R_g^k (r) + \chi_g \sum_{g'=1}^{G} \nabla \sigma_{f_{g'}} (r) \phi_{g'}^{k+1} (r) + \sum_{g' \neq g} \sigma_{s_{g'} \rightarrow g} (r) \phi_{g'}^{k+1} (r) \]

(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
The diffusion coefficient correction scheme is one such scheme. In this scheme we redefine the corrected diffusion coefficient $D_g(r)$ as

$$D_g(r) = \frac{-\tilde{f}_g(r)}{\nabla \phi_g(r)}, \quad (9)$$

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

$$-\nabla \cdot D_g^{l-1}(r) \cdot \nabla \phi^l_g(r) + \sigma_{R,g}(r) \phi^l_g(r) = Q_g(r), \quad (10)$$

and the multigroup (outer iteration) diffusion equation becomes

$$-\nabla \cdot D_g^k(r) \cdot \nabla \phi^k_g(r) + \sigma_{R,g}(r) \phi^k_g(r) = \frac{\kappa_g}{k_{eff}} \sum_{g'} G \sigma_{s,g'}(r) \phi^{k+1}_{g'}(r)$$

$$+ \sum_{g' \neq g} \sigma_{s,g' \rightarrow g}(r) \phi^{k+1}_{g'}(r), \quad (11)$$

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

$$\tilde{\sigma}_R^l(r) = \sigma_{R,g}(r) + \frac{R_g^l(r)}{\phi_g(r)}, \quad (12)$$

where $R_g^l(r)$ is defined by Eq. (6). With this modification, Eq. (5) becomes

$$-\nabla \cdot D_g(r) \nabla \phi^l_g(r) + \tilde{\sigma}_R^l(r) \phi^l_g(r) = Q_g(r)$$

and Eq. (8) becomes
\[- \nabla \cdot D_g (r) \nabla \phi_g^{k+1} (r) + \bar{\sigma}_{R,g} (r) \phi_g^{k+1} (r) \]
\[= \sum_{g' = 1}^{G} v \sigma_{f,g'} (r) \phi_{g'}^{k+1} + \sum_{g' \neq g} \sigma_{s,g' \rightarrow g} (r) \phi_{g'}^{k+1} (r). \] (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, $\nabla \cdot \Omega \psi$ or $(\Omega \cdot \nabla \psi)$ 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 $\psi (r, E, \Omega)$ is assumed independent of the azimuthal angle $\phi$ so that the angular dependence is reduced to the $\mu$ 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 $\xi$ angular cosine and also symmetric about the $\mu - \xi$, (or $\phi = 0^\circ - 180^\circ$) plane. Thus, only one-fourth of the unit sphere need be considered in the angular dependence.

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

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

$$\sigma_s (r, E \rightarrow E, \Omega' \cdot \Omega) = \sum_{L=0}^{ISCT} \left( \frac{2L+1}{4\pi} \right) \sigma_s^L (r, E \rightarrow E) P_L (\Omega' \cdot \Omega) \hspace{1cm} (14)$$

If this expansion is inserted into Eq. (1) and the addition theorem for spherical harmonics used to expand $P_n (\Omega' \cdot \Omega)$, the scattering source becomes

$$\int \int dE' d\Omega' \sigma_s (r, E \rightarrow E, \Omega' \cdot \Omega') \psi (r, E', \Omega') = SS$$

$$SS = \int dE' \sum_{L=0}^{ISCT} \left( \frac{2L+1}{4\pi} \right) \sigma_s^L (r, E' \rightarrow E) \left\{ P_L (\mu) \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' P_L (\mu') \psi (r, E', \mu', \phi') \right\}$$

$$+ 2 \sum_{K=1}^{L} \frac{(L-K)!}{(L+K)!} P^K_L (\mu) \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' P^K_L (\mu') \cos K (\phi - \phi') \psi (r, E', \mu', \phi') \right\} \right\}$$

(15)

where for cylindrical geometry we must replace the $\mu$ variable with $\xi$. Using the relation $\cos L (\phi - \phi') = \cos \phi \cos \phi' + \sin \phi \sin \phi'$, we can write Eq. (15) as

$$SS = \int dE' \sum_{L=0}^{ISCT} \frac{2L+1}{4\pi} \sigma_s^L (r, E' \rightarrow E) \left\{ P_L (\mu) \phi_L (r, E') \right\}$$

$$+ \sum_{K=1}^{L} \frac{2(L-K)!}{(L+K)!} \left[ \phi^K_{C,L} (r, E') P^K_L (\mu) \cos K \phi + \phi^K_{S,L} (r, E') P^K_L (\mu) \sin K \phi \right] \right\} \right\}$$

(16)

where we have defined the moments of the angular flux as

$$\phi_L (r, E') = \frac{1}{4\pi} \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' P_L (\mu') \psi (r, E', \mu', \phi') \hspace{1cm} (17a)$$

$$\phi^K_{C,L} (r, E') = \frac{1}{4\pi} \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' \psi (r, E', \mu', \phi') \frac{2(L-K)!}{(L+K)!} P^K_L (\mu') \cos K \phi'$$

(17b)
ONEDANT METHODS

In both standard plane and spherical geometries, due to symmetry in the azimuthal angle \( \phi \), the flux moments \( \phi_{S,L}^K \) and \( \phi_{C,L}^K \) are identically zero. In cylindrical geometry (with \( \xi, \xi' \) replacing \( \mu, \mu' \) in Eqs. (16) and (17), the odd moments \( K+L=\text{odd} \) of \( \phi_{C,L}^K \) vanish as do all the sine moments \( \phi_{S,L}^K \). In the two-angle plane geometry all moments must be retained.

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

\[
SS = \int dE \sum_{n=1}^{NM} (2L+1) \sigma_S^L (r, E \rightarrow E) R_n(\Omega) \tilde{\phi}_n(r, E),
\]

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 \( R_n(\Omega) \) are the spherical harmonics appropriate to the particular geometry, and the \( \tilde{\phi}_n(r, E) \) are the angular flux moments corresponding to the \( R_n(\Omega) \). 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 \( R_n(\Omega) \), \( 0 \leq L \leq ISCT \). The \( R_n(\Omega) \) are listed in Table 12.3 for typical Legendre expansion orders. For each \( R_n(\Omega) \) 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 $L_o$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>12</td>
<td>36</td>
</tr>
</tbody>
</table>

\[
N = \begin{cases} 
  L_o + 1 & \text{for standard plane and spherical geometry} \\
  (L_o + 2)^2 / 4 & \text{for cylindrical geometry} \\
  (L_o + 1)^2 & \text{for two-angle plane geometry}
\end{cases}
\]
Table 12.3 Spherical Harmonics, \( R_n(\Omega) \), for Different Geometries

<table>
<thead>
<tr>
<th>n</th>
<th>Standard Plane and Spherical Geometries ( P^a_n )</th>
<th>Cylindrical Geometry ( P_4 )</th>
<th>Two-Angle Plane ( P_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( P_0(\mu) )</td>
<td>( P_0(\xi) )</td>
<td>( P_0(\mu) )</td>
</tr>
<tr>
<td>2</td>
<td>( P_1(\mu) )</td>
<td>( P_1(\xi) \cos \phi )</td>
<td>( P_1(\mu) )</td>
</tr>
<tr>
<td>3</td>
<td>( P_2(\mu) )</td>
<td>( P_2(\xi) )</td>
<td>( P_1^1(\mu) \cos \phi )</td>
</tr>
<tr>
<td>4</td>
<td>( P_3(\mu) )</td>
<td>( \frac{\sqrt{3}}{2} P_2^2(\xi) \cos 2\phi )</td>
<td>( P_1(\mu) \sin \phi )</td>
</tr>
<tr>
<td>5</td>
<td>( P_4(\mu) )</td>
<td>( \frac{\sqrt{6}}{6} P_3^1(\xi) \cos 2\phi )</td>
<td>( P_2(\mu) )</td>
</tr>
<tr>
<td>6</td>
<td>( P_5(\mu) )</td>
<td>( \frac{\sqrt{10}}{60} P_3^3(\xi) \cos 3\phi )</td>
<td>( \frac{\sqrt{3}}{3} P_2^1(\mu) \cos \phi )</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>( P_4(\xi) )</td>
<td>( \frac{\sqrt{3}}{3} P_2^1(\mu) \sin \phi )</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>( \frac{\sqrt{5}}{30} P_4^2(\xi) \cos 2\phi )</td>
<td>( \frac{\sqrt{6}}{6} P_2^2(\mu) \cos 2\phi )</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>( \frac{\sqrt{75}}{840} P_4^4(\xi) \cos 4\phi )</td>
<td>( \frac{\sqrt{5}}{6} P_2^2(\mu) \sin 2\phi )</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>( P_3(\mu) )</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>( \frac{\sqrt{6}}{6} P_3^1(\mu) \cos \phi )</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>( \frac{\sqrt{6}}{6} P_3^1(\mu) \sin \phi )</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>( \frac{\sqrt{15}}{30} P_3^2(\mu) \cos 2\phi )</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>( \frac{\sqrt{15}}{30} P_3^2(\mu) \sin 2\phi )</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>( \frac{\sqrt{10}}{60} P_3^3(\mu) \cos 3\phi )</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>( \frac{\sqrt{10}}{60} P_3^3(\mu) \sin 3\phi )</td>
<td></td>
</tr>
</tbody>
</table>

a. \( P_n \) 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 \( Q(r, E, \Omega) \) can be represented as a finite expansion using the spherical harmonics \( R_n(\Omega) \) defined in Table 12.3. First, the inhomogeneous source moments are defined for a Legendre expansion order \( IQAN \):

\[
Q_L(r, E) = \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi Q(r, E, \Omega) P_L(\mu), \quad L = 0, ..., IQAN, \quad (19a)
\]

\[
Q^K_{C,L}(r, E) = \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi Q(r, E, \Omega) P^K_{L}(\mu) \cos K\phi, \quad (19b)
\]

\[
Q^K_{S,L}(r, E) = \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi Q(r, E, \Omega) P^K_{L}(\mu) \sin K\phi, \quad (19c)
\]

The inhomogeneous source is represented in the general spherical harmonic expansion

\[
Q(r, E, \Omega) = \sum_{n=1}^{NMQ} (2L + 1) R_n(\Omega) \tilde{Q}_n(r, E), \quad (20)
\]

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 \( R_n(\Omega) \) are the spherical harmonics appropriate to the geometry being used, and the \( \tilde{Q}_n(r, E) \) are the angular source moments corresponding the \( R_n(\Omega) \). \( IQAN \) is determined from the number of spherical harmonics, \( NMQ \), in the input. The index \( L \) is the subscript of the Legendre function \( P^K_L \) appearing in the appropriate \( R_n(\Omega) \), \( 0 \leq L \leq IQAN \). The \( R_n(\Omega) \) are listed in Table 12.3 for typical Legendre expansion orders. For each of these \( R_n(\Omega) \) 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, $N$

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>24</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>48</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>80</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>168</td>
<td>42</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>288</td>
<td>72</td>
<td>16</td>
</tr>
<tr>
<td>$N$</td>
<td>$N$</td>
<td>$N \cdot (N + 2)$</td>
<td>$N \cdot (N + 2)/4$</td>
<td>$N$</td>
</tr>
</tbody>
</table>

a. Standard Plane Geometry

For standard plane geometry (see Table 12.1 and Figure 12.1) azimuthal symmetry is assumed in $\phi$ so that $\Omega (\mu, \phi)$ becomes $\Omega (\mu)$ and $d\Omega$ becomes $2\pi d\mu$. The angular interval $\mu \in [-1,1]$ is discretized into $MM$ quadrature points $\mu_m$ 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

$$\phi_L (x) \equiv \sum_{m=1}^{MM} w_m P_L (\mu_m) \psi_m (x), \quad (21)$$

The discrete-ordinates approximation to the transport equation, i.e., Eq. (3), becomes

$$\mu_m \frac{\partial \psi_m (x)}{\partial x} + \sigma (x) \psi_m (x) = S_m (x) \quad (22)$$
Figure 12.4  Ordering of S₆ 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 \((\mu_m, \phi_m)\) 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

\[
\phi_L(x) \equiv \sum_{m=1}^{MM} w_m P_L(\mu_m) \psi_m(x),
\]

(23a)

\[
\phi_{C,L}^K(x) \equiv \sum_{m=1}^{MM} w_m \psi_m(x) P_L^K(\mu_m) \cos K\phi_m,
\]

(23b)

\[
\phi_{S,L}^K(x) = \sum_{m=1}^{MM} w_m \psi_m(x) P_L^K(\mu_m) \sin K\phi_m.
\]

(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.
c. Cylindrical Geometry

For cylindrical geometry (see Table 12.1 and Figure 12.2), the multigroup transport equation, Eq. (3), may be written

$$
\mu \frac{\partial}{\partial r} (r \psi) - \frac{\partial (r \sigma \psi)}{\partial \phi} + r \sigma \psi = r \lambda (r, \Omega),
$$

(24)

where $\psi = \psi (r, \Omega)$.

For the discrete-ordinates approximation in cylindrical geometry, only one quadrant of the unit sphere is discretized into a set of MM quadrature points $(\mu_m, \eta_m)$ 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](image)

As before, $\psi_m(r) \equiv \psi(r, \mu_m, \eta_m)$ represents the average angular flux in $d\Omega_m$ about $\Omega_m$ 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 $\xi$-level as $\psi_{m - 1/2}(r)$ and $\psi_{m + 1/2}(r)$. The discrete-ordinates approximation to Eq. (24) can then be written:

$$\mu_m \frac{\partial (r \psi_m)}{\partial r} + \left( \frac{\alpha_{m + 1/2}}{w_m} \right) \psi_{m + 1/2}(r) - \left( \frac{\alpha_{m - 1/2}}{w_m} \right) \psi_{m - 1/2}(r) + r \sigma \psi_m(r)$$

where the $\alpha_{m - 1/2}$ and $\alpha_{m + 1/2}$ are angular coupling coefficients. These coefficients satisfy the recursion relation.
\( \alpha_{m+1/2} - \alpha_{m-1/2} = -w_m \mu_m, \) \hspace{1cm} (26)

with the requirement that the first \((\alpha_{1/2})\) and last \((\alpha_{M+1/2})\) coefficients on each \(\xi\)-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 \(\left(\frac{\sigma_{m+1/2}}{w_m}\right)\) 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

\[
\mu \frac{\partial (r^2 \psi)}{\partial r} + r \frac{\partial [(1 - \mu^2) \psi]}{\partial \mu} + r^2 \sigma \psi = r^2 S(r, \mu),
\] \hspace{1cm} (27)

where azimuthal symmetry in \(\phi\) (see Figure 12.3) has been assumed. The angular domain \(\mu \in [-1, 1]\) is discretized into \(MM\) quadrature points \(\mu_m\) 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 \(\Omega_m\) 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

\[
\mu_m \frac{\partial (r^2 \psi_m)}{\partial r} + \left[ \left( \frac{\beta_{m+1/2}}{w_m} \right) \psi_{m+1/2}(r) - \left( \frac{\beta_{m-1/2}}{w_m} \right) \psi_{m-1/2}(r) \right] r + r^2 \sigma \psi_m(r) = r^2 S_m(r),
\] \hspace{1cm} (28)

where the angular coupling coefficients \(\beta\) must satisfy the recursion relation

\[
\beta_{m+1/2} - \beta_{m-1/2} = -2w_m \mu_m, \quad m = 1, ..., MM,
\] \hspace{1cm} (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 ONEDANT...
ANT pertaining to the angular quadrature, the quantities \( \frac{(\beta_{m+1/2})}{2w_m} \) and \( \frac{(\beta_{m-1/2})}{2w_m} \) 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 \( \psi_m \), namely,

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

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, \( \eta = 0, \varphi = 180^\circ \), are used for each \( \xi \)-level to yield special equations for \( \psi_{1/2} (r) \) on each \( \xi \)-level. At the origin, \( r=0 \), we also impose an isotropic condition on the angular flux; i.e., \( \frac{\partial \psi}{\partial \mu} \bigg|_{r=0} = 0 \). This implies that \( \psi_m (0) = \) constant for all \( m \). The constant is determined by solving the starting direction equation as \( \frac{d \psi}{d r} + \sigma \psi = S \) which applies to spherical geometry and to cylindrical geometry on each \( \xi \) level.

**Discretization of the Spatial Variable**

The spatial domain of the problem is ultimately partitioned into \( \text{IT} \) fine-mesh intervals of width \( \Delta x_i, i = 1, 2, ..., \text{IT} \) such that \( \Delta x_i = x_{i+1/2} - x_{i-1/2} \). 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:

\[
\mu_m \left( A_{i+1/2} \psi_{g,m,i+1/2} - A_{i-1/2} \psi_{g,m,i-1/2} \right) + \\
+ (A_{i+1/2} - A_{i-1/2}) \left( \alpha_{m+1/2} \psi_{g,m+1/2,i} - \alpha_{m-1/2} \psi_{g,m-1/2,i} \right) \psi_m + \\
+ \sigma_{t,g,i} \psi_{g,m,i} = S_{g,m,i} V_i 
\]

\( m = 1,\ldots,MM; \quad i = 1,\ldots,IT \)  \hspace{1cm} (31)

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:

\[
\psi_{g,m,i} = 0.5 \left( \psi_{g,m,i+1/2} + \psi_{g,m,i-1/2} \right) \\
\psi_{g,m,i} = 0.5 \left( \psi_{g,m+1/2,i} + \psi_{g,m-1/2,i} \right) 
\]

\( m = 1,\ldots,M; \quad i = 1,\ldots,IT \)  \hspace{1cm} (32)

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

<table>
<thead>
<tr>
<th>n</th>
<th>Two-Dimensional Geometries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$P_0(\mu)$</td>
</tr>
<tr>
<td>2</td>
<td>$P_1(\mu)$</td>
</tr>
<tr>
<td>3</td>
<td>$P_1^1(\mu) \cos \phi$</td>
</tr>
<tr>
<td>4</td>
<td>$P_2(\mu)$</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{\sqrt{3}}{3} P_2^1(\mu) \cos \phi$</td>
</tr>
<tr>
<td>6</td>
<td>$\frac{\sqrt{3}}{6} P_2^2(\mu) \cos 2\phi$</td>
</tr>
<tr>
<td>7</td>
<td>$P_3(\mu)$</td>
</tr>
<tr>
<td>8</td>
<td>$\frac{\sqrt{6}}{6} P_3^1(\mu) \cos \phi$</td>
</tr>
<tr>
<td>9</td>
<td>$\frac{\sqrt{15}}{30} P_3^2(\mu) \cos 2\phi$</td>
</tr>
<tr>
<td>10</td>
<td>$\frac{\sqrt{10}}{60} P_3^3(\mu) \cos 3\phi$</td>
</tr>
</tbody>
</table>
This table shows the specific case for P₃ 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

<table>
<thead>
<tr>
<th>(N)</th>
<th>Triangular Arrangement</th>
<th>Square Arrangement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>36</td>
</tr>
<tr>
<td>16</td>
<td>36</td>
<td>64</td>
</tr>
<tr>
<td>(N)</td>
<td>(N(N+2)/8)</td>
<td>(N\cdot N/4)</td>
</tr>
</tbody>
</table>

### 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**

\[
\Omega_m \cdot \nabla \psi_m = \mu_m \frac{\partial \psi_m}{\partial x} + \eta_m \frac{\partial \psi_m}{\partial y}
\]

where,

\[
\mu_m = \hat{e}_x \cdot \Omega_m,
\]

\[
\eta_m = \hat{e}_y \cdot \Omega_m.
\]
Planar $R-\Theta$ Symmetry

\[ \mathbf{\Omega}_m \cdot \nabla \psi_m = \frac{\mu_m}{r} \frac{\partial}{\partial r} (r \psi_m) + \frac{\eta_m}{r} \frac{\partial \psi_m}{\partial \theta} - \frac{1}{r} \frac{\partial}{\partial \phi} (\eta \psi_m) \]  

(34)

where,

\[ \mu_m = \hat{e}_r \cdot \mathbf{\Omega}_m, \]

\[ \eta_m = \hat{e}_\theta \cdot \mathbf{\Omega}_m. \]

Cylindrical $R-Z$ Symmetry

\[ \mathbf{\Omega}_m \cdot \nabla \psi_m = \frac{\mu_m}{r} \frac{\partial}{\partial r} (r \psi_m) - \frac{1}{r} \frac{\partial}{\partial \phi} (\eta \psi_m) + \xi_m \frac{\partial \psi_m}{\partial z} \]  

(35)

where,

\[ \mu_m = \hat{e}_r \cdot \mathbf{\Omega}_m, \]

\[ \xi_m = \hat{e}_z \cdot \mathbf{\Omega}_m. \]

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.

\[ \mu_m \left( A_{i+1/2,j} \psi_{g,m,i+1/2,j} - A_{i-1/2,j} \psi_{g,m,i-1/2,j} \right) + \]

\[ + \eta_m B_{i,j} \left( \psi_{g,m,i,j+1/2} - \psi_{g,m,i,j-1/2} \right) + \]

\[ + (A_{i+1/2,j} - A_{i-1/2,j}) \left( \frac{\alpha_m + 1/2 \psi_{g,m+1/2,i,j} - \alpha_m - 1/2 \psi_{g,m-1/2,i,j}}{\psi_m} \right) + \]

\[ + \sigma_{t,g,i,j} \psi_{g,m,i,j} \psi_{g,m,i,j} = S_{g,m,i,j} \]  

(36)

where,

\[ \psi_{g,m,i+1/2,j} \text{ is the flux on the right edge of the mesh cell,} \]

\[ \psi_{g,m,i,j+1/2} \text{ is the flux on the top edge of the mesh cell,} \]
\( \psi_{g,m+1/2,i,j} \) 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:

\[
\begin{align*}
\psi_{g,m,i,j} &= 0.5 (\psi_{g,m,i+1/2,j} + \psi_{g,m,i-1/2,j}) \\
\psi_{g,m,i,j} &= 0.5 (\psi_{g,m,i,j+1/2} + \psi_{g,m,i,j-1/2}) \\
\psi_{g,m,i,j} &= 0.5 (\psi_{g,m+1/2,i,j} + \psi_{g,m-1/2,i,j})
\end{align*}
\]

(37)

\( m = 1,...,MM; \quad i = 1,...,IT; \quad j=1,...,JT \)

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:
TWODANT METHODS

Spatially Discretized Two-Dimensional Transport Equation

\[
(1 + P_{x,g,m,i,j}) \psi_{g,m,i,j} = \left( \frac{\psi_{g,m,i+1/2,j} + P_{x,g,m,i,j} \psi_{g,m,i-1/2,j}}{P_{x,g,m,i,j}} \right) \left( \begin{array}{c} \mu_m > 0 \\ \mu_m < 0 \end{array} \right)
\]

\[
(1 + P_{y,g,m,i,j}) \psi_{g,m,i,j} = \left( \frac{\psi_{g,m,i,j+1/2} + P_{y,g,m,i,j} \psi_{g,m,i,j-1/2}}{P_{y,g,m,i,j}} \right) \left( \begin{array}{c} \eta_m > 0 \\ \eta_m < 0 \end{array} \right)
\]

\[
(1 + P_{a,g,m,i,j}) \psi_{g,m,i,j} = P_{a,g,m,i,j} \psi_{g,m,i,j+1/2} - \psi_{g,m,i,j-1/2, i,j} \]

\[
m = 1,...,MM; \quad i = 1,...,IT; \quad j = 1,...,JT
\]

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 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. 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. This capability includes multigroup problems involving upscatter and fission.

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 MCBND, MCJBND, MCREG, and MCBLT control the size and location of the Monte Carlo region. MCBND designates the left and right boundaries of the Monte Carlo region, where the numerical values represent an $S_n$ fine mesh cell. Thus, if IT=3, MCBND=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 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 ALPHAN(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 out 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, then 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 ($w$/...
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 \( R_{ij} - \text{Int} R_{ij} \), \( \text{Int} R_{ij} + 1 \) particles are created; otherwise, \( \text{Int} R_{ij} \) 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 \( \eta \). 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 \( \eta \) 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 \( \eta \) 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 \( \eta \) 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 \( \sigma_f/\sigma_i \), 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 fissile materials, where the weight multiplier is now \((\sigma_s + \sigma_f)/\sigma_i \). 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 \( w_m - \text{Int} w_m \), it creates \( \text{Int} w_m + 1 \) particles; otherwise, \( \text{Int} w_m \) 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.
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.
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.
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


ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT – Code Structure

Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory
# TABLE OF CONTENTS

**TABLE OF CONTENTS**........................................................................................................ 13-3

**LIST OF TABLES**........................................................................................................ 13-5

**OPERATION OF THE CODE SYSTEM**........................................................................ 13-7
  - Programming Practices and Standards ....................................................................... 13-7
  - Language ....................................................................................................................... 13-7
  - Structure ....................................................................................................................... 13-7
  - Standard Interface Files ............................................................................................... 13-7
  - Data Management and Transfers ............................................................................... 13-8
  - Central Memory Restrictions ....................................................................................... 13-8
  - Word Size ..................................................................................................................... 13-8

**CODE PACKAGE STRUCTURE**....................................................................................... 13-9
  - Input Module.................................................................................................................. 13-16
  - Solver Modules ............................................................................................................ 13-17
  - Edit Module .................................................................................................................. 13-17

**PIECEWISE EXECUTION**.................................................................................................. 13-19
  - Module Execution Control ........................................................................................... 13-19
  - Submodule Execution Control (File Generation Suppression) ....................................... 13-20

**STACKED RUNS**............................................................................................................ 13-25

**REFERENCES**............................................................................................................... 13-27

---

Code Structure Version 3.0 13-3
LIST OF TABLES

Table 13.1: Files Read and Written ........................................................................... 13-9
Table 13.2: Structure of the Input Module .............................................................. 13-12
Table 13.3: Structure of the Onedant Solver Module ........................................... 13-13
Table 13.4: Structure of the Twodant Solver Module .......................................... 13-13
Table 13.5: Structure of the Threedant Solver Module ......................................... 13-14
Table 13.6: Structure of the Edit Module .............................................................. 13-15
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.)
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>
<thead>
<tr>
<th>Information Type</th>
<th>File</th>
<th>Input Module</th>
<th>Solver Module</th>
<th>Edit Module</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Read</td>
<td>Write</td>
<td>Read</td>
</tr>
<tr>
<td>Geometry Information</td>
<td>GEODST</td>
<td>O</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Card Images</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GEOSING²</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 13.1 Files Read and Written
<table>
<thead>
<tr>
<th>Information Type</th>
<th>File</th>
<th>Input Module</th>
<th>Solver Module</th>
<th>Edit Module</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Read</td>
<td>Write</td>
<td>Read</td>
</tr>
<tr>
<td>Cross Sections</td>
<td>ISOTXS</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GRUPXS</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MENDEF</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MACRXS / ADJMAC</td>
<td>O</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>MACBCD</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XSLIB</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>XSLIBE</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>XSLIBF</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BXSLIB</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>XSLIBB</td>
<td>O</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SNXEDT</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Card Image</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material Mixing</td>
<td>NDXSRF/ZNATDN</td>
<td>O</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Card Images</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LNK3DNTb</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment of Materials to Zones</td>
<td>ASGMAT</td>
<td>A</td>
<td>A</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>Card Images</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solver Module Input</td>
<td>SOLINP</td>
<td>O</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>Card Images</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quadrature</td>
<td>SNCONS</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inhomogeneous Sources</td>
<td>FIXSRC</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Card Images</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Module Input</td>
<td>EDITIT</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Card Images</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information Type</td>
<td>File</td>
<td>Input Module</td>
<td>Solver Module</td>
<td>Edit Module</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Read</td>
<td>Write</td>
<td>Read</td>
</tr>
<tr>
<td>Other Output Files</td>
<td>RTFLUX/ATFLUX</td>
<td></td>
<td></td>
<td>O&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>RAFLUX/AAFLUX</td>
<td></td>
<td></td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>RZFLUX/AZFLUX</td>
<td></td>
<td></td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>RMFLUX/AMFLUX</td>
<td></td>
<td></td>
<td>O&lt;sup&gt;d&lt;/sup&gt;</td>
</tr>
<tr>
<td></td>
<td>RAFLXM/AAFLXM</td>
<td></td>
<td></td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>RZMFLX/AZMFLX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ARBFLUX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FISSRC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EDTOUT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EDTOGX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>UCFLUX</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPT10</td>
<td>Input Module Driver: controls the flow of the code by calling one or more of the secondary somebodies below.</td>
</tr>
<tr>
<td>INPT11</td>
<td>Controls code setup and storage allocation.</td>
</tr>
<tr>
<td>INPT12</td>
<td>Controls geometry data processing</td>
</tr>
<tr>
<td>INPT13</td>
<td>Controls cross-section library processing for XSLIB, MENDF, XSLIBB, and MACBCD library forms.</td>
</tr>
<tr>
<td>INPT14</td>
<td>Controls mixing specification processing.</td>
</tr>
<tr>
<td>INPT15</td>
<td>Controls GRUPXS cross-section library processing.</td>
</tr>
<tr>
<td>INPT16</td>
<td>Controls ISOTXS cross-section library processing.</td>
</tr>
<tr>
<td>INPT17</td>
<td>Controls BXSLIB cross-section library processing.</td>
</tr>
<tr>
<td>INPT18</td>
<td>Controls Solver Module input data processing.</td>
</tr>
<tr>
<td>INPT19</td>
<td>Controls Edit Module input data processing.</td>
</tr>
<tr>
<td>INPT110</td>
<td>Controls cross-section balancing operation.</td>
</tr>
<tr>
<td>INPT111</td>
<td>Controls adjoint reversals.</td>
</tr>
<tr>
<td>INPT112</td>
<td>Controls GEODST file post processing.</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRND20</td>
<td>Solver Module Driver; controls the flow of the code by calling one or more of the secondary submodules below.</td>
</tr>
<tr>
<td>INPT21</td>
<td>Controls module initializations.</td>
</tr>
<tr>
<td>INPT22</td>
<td>Controls quadrature selection.</td>
</tr>
<tr>
<td>INPT23</td>
<td>Controls flux guess and inhomogeneous source processing.</td>
</tr>
<tr>
<td>GRND24</td>
<td>Controls calculational data preparation</td>
</tr>
<tr>
<td>GRND25</td>
<td>Controls the outer iterations.</td>
</tr>
<tr>
<td>OUTT26</td>
<td>Controls final Solver Module printing.</td>
</tr>
<tr>
<td>OUT27</td>
<td>Controls binary file preparation.</td>
</tr>
</tbody>
</table>

**Table 13.4 Structure of the Twodant Solver Module**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIGF20</td>
<td>Solver Module Driver; controls the flow of the code by calling one or more of the secondary submodules below.</td>
</tr>
<tr>
<td>TINP21</td>
<td>Controls module initializations.</td>
</tr>
<tr>
<td>TINP22</td>
<td>Controls flux guess and inhomogeneous source processing.</td>
</tr>
<tr>
<td>TINP23</td>
<td>Controls quadrature selection.</td>
</tr>
<tr>
<td>TINP24</td>
<td>Checks spatial mesh input for consistency.</td>
</tr>
<tr>
<td>TGND25</td>
<td>Calculates initially required functions and grid structures; calculates the first collision source and Monte Carlo Source if required, initializes fission source.</td>
</tr>
<tr>
<td>TRANSO</td>
<td>Controls the inner iterations.</td>
</tr>
</tbody>
</table>
Table 13.4 Structure of the Twodant Solver Module (Cont.)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFFO</td>
<td>Controls the outer iterations.</td>
</tr>
<tr>
<td>LINKO</td>
<td>Controls the $S_n$/MC iterations.</td>
</tr>
<tr>
<td>TOT28</td>
<td>Controls final Solver Module printing.</td>
</tr>
<tr>
<td>TOT29</td>
<td>Controls binary file preparation.</td>
</tr>
</tbody>
</table>

Table 13.5 Structure of the Threedant Solver Module

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIGFA03D</td>
<td>Solver Module Driver; controls the flow of the code by calling one or more of the secondary submodules below.</td>
</tr>
<tr>
<td>TINP213D</td>
<td>Controls module initializations.</td>
</tr>
<tr>
<td>TINP223D</td>
<td>Controls flux guess and inhomogeneous source processing.</td>
</tr>
<tr>
<td>TINP233D</td>
<td>Controls quadrature selection.</td>
</tr>
<tr>
<td>TINP243D</td>
<td>Checks spatial mesh input for consistency.</td>
</tr>
<tr>
<td>TINP253D</td>
<td>Calculates initially required functions and grid structures; initializes the fission source.</td>
</tr>
<tr>
<td>TRANSO3D</td>
<td>Controls the inner iteration.</td>
</tr>
<tr>
<td>DIFFO3D</td>
<td>Controls the outer iteration.</td>
</tr>
<tr>
<td>TOT283D</td>
<td>Controls final Solver Module printing.</td>
</tr>
<tr>
<td>TOT293D</td>
<td>Controls binary file preparation.</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>Routine</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTT30</td>
<td>Edit module driver; controls the flow of the code by calling one or more of the secondary submodules below.</td>
</tr>
<tr>
<td>OUTT31</td>
<td>Controls reaction rate calculations.</td>
</tr>
<tr>
<td>OUTT32</td>
<td>Controls power normalization, edit zone averaging, and output file preparation.</td>
</tr>
<tr>
<td>OUTT33</td>
<td>Controls a spatial mesh collapse determination.</td>
</tr>
<tr>
<td>OUTT34</td>
<td>Controls the mass edit request on the coarse mesh or edit zones.</td>
</tr>
</tbody>
</table>

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 discreteordinates approximation is used for treating the angular variation of the particle distribution and the diamond-difference scheme or the adaptive weighted diamond difference scheme in 2 and 3-d, 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, 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 $\sigma$ is a particular multigroup, microscopic cross section for a particular isotope or nuclide and $\phi$ 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.
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 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 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.


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.


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.”


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.
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


TABLE OF CONTENTS

TABLE OF CONTENTS ............................................................................................ 14-3

INPUT ERRORS ........................................................................................................ 14-5
  Examples of Errors and Resulting Messages ........................................................ 14-5
  Sample Error 1. Misspelled Input Array Name ...................................................... 14-5
  Sample Error 2. Input Block Terminator Omitted ............................................... 14-6
  Sample Error 3. Invalid Entry for Block-I Input Parameter. ............................... 14-7
  Sample Error 4. Incorrect Number of Entries in an Input Array. ....................... 14-7
  Sample Error 5. Misplaced Array Identifier ....................................................... 14-8

COMMENTS REGARDING MULTIPLE ERRORS .............................................. 14-11

IMPLEMENTATION ERRORS .............................................................................. 14-13

WARNINGS ............................................................................................................ 14-15
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 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
***error** current block contains arrays belonging to other blocks
```

```
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

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 *)
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

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

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.
COMMENTS REGARDING MULTIPLE 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.
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.
# TABLE OF CONTENTS

## TABLE OF CONTENTS

- TABLE OF CONTENTS ................................................................. 15-3
- INTRODUCTION ................................................................. 15-5

## ASCII FILES

- Problem Input File ................................................................. 15-7
- Cross-Section Library Files .................................................. 15-7
- EDTOUT ................................................................. 15-7
- EDTOGX ................................................................. 15-13
- ARBFLUX ................................................................. 15-17

## STANDARD INTERFACE FILES

- STANDARD INTERFACE FILES .............................................. 15-19

## CODE DEPENDENT INTERFACE FILES

- CODE DEPENDENT INTERFACE FILES .................................... 15-21
  - AAFLXM for TWODANT ...................................................... 15-22
  - AAFLXM for THREEDANT .................................................. 15-24
  - ADJMAC ................................................................. 15-28
  - AMFLUX ................................................................. 15-31
  - ASGMA  ................................................................. 15-33
  - AZMFLX ................................................................. 15-36
  - BXSLIB ................................................................. 15-38
  - EDITIT ................................................................. 15-40
  - FISSRC ................................................................. 15-47
  - GEOODST ............................................................... 15-49
  - GEOSEING ............................................................... 15-61
  - LNK3DNT ................................................................. 15-62
  - MACRXX ................................................................. 15-64
  - RAFLXM for TWODANT .................................................. 15-67
  - RAFLXM for THREEDANT ................................................ 15-70
  - RMFLUX ................................................................. 15-74
  - RZMFLX ................................................................. 15-76
  - SNXEDT ................................................................. 15-78
  - SOLINP ................................................................. 15-81
  - UCFLUX ................................................................. 15-96

## OVERWRITTEN INPUT FILES

- OVERWRITTEN INPUT FILES ................................................ 15-99

## REFERENCES

- REFERENCES ................................................................. 15-101

---

FILE DESCRIPTIONS Version 3.0 15-3
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.
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 (IMPl), 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 (JMPl), 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 (KMPl), 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)]
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)
   CONTINUE
   CONTINUE
   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)
   CONTINUE
   CONTINUE
   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

FILE DESCRIPTIONS Version 3.0
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.
1. 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 (JMPl), 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)
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.
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.
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.
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.

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.

-----------------------------------------------------------------------

FILE STRUCTURE

FILE IDENTIFICATION

* RECORD TYPE PRESENT IF
* FILE IDENTIFICATION ALWAYS
* FILE CONTROL ALWAYS

*(REPEAT FOR ALL GROUPS)
* (GROUP 1 IS FIRST)
* *(REPEAT FOR ALL DIRECTIONS)
* * COMPUTATIONAL COSINES ALWAYS
* * HORIZONTAL-EDGE ANGULAR FLUX ALWAYS
* * VERTICAL-EDGE ANGULAR FLUX ALWAYS

FILE IDENTIFICATION

HNAME,(HUSE(I),I=1,2),IVERS
1+3*MULT=NUMBER OF WORDS
HNAME HOLLERITH FILE NAME - AAFLUX - (A6)
HUSE(I) HOLLERITH USER IDENTIFICATION (A6)
IVERS FILE VERSION NUMBER
MULT DOUBLE PRECISION PARAMETER
1- A6 WORD IS SINGLE WORD
2- A6 WORD IS DOUBLE PRECISION WORD
CODE DEPENDENT INTERFACE FILES

CR  SPECIFICATIONS  (1D RECORD)

CL  NDIM,NGROUP,NINTI,NINTJ,NINTK,NDIR,EFFK,ADUM

CD  8=NUMBER OF WORDS

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  NINTK  NUMBER OF THIRD DIMENSION FINE MESH INTERVALS
CD  NINTJ.EQ.1 IF NDIM.EQ.1
CD  NINTK.EQ.1 IF NDIM.LE.2
CD  NDIR  NUMBER OF DIRECTIONS
CD  EFFK  EFFECTIVE MULTIPLICATION FACTOR
CD  ADUM  RESERVED

CR  HORIZONTAL EDGE ADJOINT ANGULAR FLUX

CL  ((HEDGE(I,J),I=NBDRYI),J=1,NINTJ)

CD  NBDRYI*NINTJ*MULTI=NUMBER OF WORDS
CD  HEDGE(I,J)  HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX
CD  NBDRYI  NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH
          BOUNDARIES)

CR  VERTICAL EDGE ADJOINT ANGULAR FLUX

CL  ((VEDGE(I,J),I=NINTI),J=1,NBDRYJ)

CD  NINTI*NBDRYJ*MULTI=NUMBER OF WORDS
CD  VEDGE(I,J)  HORIZONTAL-EDGE-BOUNDARY ANGULAR FLUX
CD  NBDRYJ  NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH
          BOUNDARIES)

VERITCAL EDGE
--------------
|               |
|               |
HORIZONTAL EDGE
--------------

CEOF

FILE DESCRIPTIONS  Version 3.0  15-23
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
DATE 3/03/95

THIS FILE PROVIDES THE EDGE ANGULAR FLUX AS ORDERED BY THE CODE

ORDER OF GROUPS IS ACCORDING TO INCREASING ENERGY. NOTE THAT DOUBLE PRECISION FLUXES ARE GIVEN WHEN MULT=2
```

FILE STRUCTURE

```c
RECORD TYPE PRESENT IF

FILE IDENTIFICATION ALWAYS
FILE CONTROL ALWAYS

**********(REPEAT FOR ALL GROUPS)
* (GROUP NGROUP IS FIRST)
* **********(REPEAT FOR ALL FRONT-GOING DIRECTIONS)
* * COMPUTATIONAL COSINES ALWAYS
* * CELL BACK-EDGE ANGULAR FLUX FOR BACK PLANE
* *
* **********(REPEAT FOR ALL Z-INTERVALS)
* (INTERVAL NINTK IS FIRST)
* * **********(REPEAT FOR ALL FRONT-GOING DIRECTIONS)
* * * COMPUTATIONAL COSINES ALWAYS
* * * HORIZONTAL-EDGE ANGULAR FLUX ALWAYS
* * * VERTICAL-EDGE ANGULAR FLUX ALWAYS
* * * CELL FRONT-EDGE ANGULAR FLUX ALWAYS
* *
```
**CODE DEPENDENT INTERFACE FILES**

**AAFLXM for THREEDANT**

---

**FILE IDENTIFICATION**

CR

CL

HNAME, (HUSE(I), I=1, 2), IVERS

CW

1 + 3 * MULT = NUMBER OF WORDS

CD

HNAME HOLLERITH FILE NAME - RAFLXM - (A6)

HUSE(I) HOLLERITH USER IDENTIFICATION (A6)

IVER FILE VERSION NUMBER

MULT DOUBLE PRECISION PARAMETER

1- A6 WORD IS SINGLE WORD

2- A6 WORD IS DOUBLE PRECISION WORD

---

**SPECIFICATIONS (1D RECORD)**

CR

CL

NDIM, NGROUP, NINTI, NINTJ, NINTK, EFFK, POWER

CW

8 = NUMBER OF WORDS

CD

NDIM NUMBER OF DIMENSIONS

NGROUP NUMBER OF ENERGY GROUPS

NINTI NUMBER OF FIRST DIMENSION FINE MESH INTERVALS

NINTJ NUMBER OF SECOND DIMENSION FINE MESH INTERVALS

NINTK NUMBER OF THIRD DIMENSION FINE MESH INTERVALS

EFFK EFFECTIVE MULTIPLICATION FACTOR

POWER POWER IN WATTS TO WHICH FLUX IS NORMALIZED

---

**COMPUTATIONAL COSINES**

CR

CL

XMU, XETA, XI, WEIGHT, M, K

CW

6 * MULT

CD

XMU MU COSINE

XETA ETA COSINE

XI XI COSINE

WEIGHT WEIGHT

---

FILE DESCRIPTIONGS Version 3.0 15-25
CODE DEPENDENT INTERFACE FILES

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-----------------------------------------------------------------------
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 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*NBDRYJ*MULT=NUMBER OF WORDS
C
CD NBDRYJ NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH 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-----------------------------------------------------------------------
CODE DEPENDENT INTERFACE FILES

AAFLXM for THREEDANT

FILE DESCRIPTIONS Version 3.0
The ADJMAC file is the adjoint-reversed counterpart to the MACRXS interface file.

```plaintext
FILE IDENTIFICATION

HNAME, (HUSE(I), I=1,2), IVERS

1+3*MULT=NUMBER OF WORDS

HNAME HOLLERITH FILE NAME - ADJMAC - (A6)
HUSE(I) HOLLERITH USER IDENTIFICATION (A6)
IVERs FILE VERSION NUMBER
MULT DOUBLE PRECISION PARAMETER

1- A6 WORD IS SINGLE WORD
2- A6 WORD IS DOUBLE PRECISION WORD

FILE CONTROL

NGROUP, NMAT, NORD, NED, IDPF, LNG, MAXUP, MAXDN, NPRIN, I2LP1

10=NUMBER OF WORDS

NGROUP NUMBER OF ENERGY GROUPS IN FILE
NMAT NUMBER OF MATERIALS IN FILE
```
CODE DEPENDENT INTERFACE FILES

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  MAXD  MAXIMUM NUMBER OF DOWNSCATTER GROUPS
CD  NFRIN NUMBER OF PRINCIPAL CROSS SECTIONS
CD  I2LP  0/1 NO/YES 2L+1 TERM WAS INCLUDED IN LIBRARY

_____________________________________________________________

FILE DATA

(CL (HMAT(I),I=1,NMAT),(HED(J),J=1,NEDT),(VEL(N),N=1,NGROUP),
 CL 1(EMAX(N),N=1,NGROUP),EMIN
 CW (NMAT+NEDT+2*NGROUP+1)*MUL=NUMBER OF WORDS
 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 EMN   MINIMUM ENERGY BOUND OF SET (EV)
 CD NEDT  NED+NFRIN

THE FOUR BASIC PRINCIPAL CROSS SECTIONS
ALWAYS PRESENT ARE:

HED(1) = 3CHIC
HED(2) = 6HNSIGF
HED(3) = 5HTOTAL
HED(4) = 3HABS

ALSO PRESENT WHEN NPRIN=5 IS:

HED(5) = 5HTRANS

PRINCIPAL CROSS SECTIONS FOR GROUP N

(CL (C(I,J),I=1,NMAT),J=1,NEDT)
 CW NMAT*NEDT*MULT=NUMBER OF WORDS
 CD C(I,J) PRINCIPAL CROSS SECTIONS

BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:

J=1 FISSION SPECTRUM
J=2 FISSION NF FISSION CROSS SECTION
J=3 TOTAL CROSS SECTION
J=4 ABSORPTION CROSS SECTION

ALSO PRESENT WHEN NPRIN=5 IS:

J=5 TRANSPORT CROSS SECTION

SCATTERING CONTROL BLOCK FOR GROUP N
PRESENT IF NORD.GT.0

((NGPB(L,J),L=1,NORD),J=1,NMAT)

((IFSG(L,J),L=1,NORD),J=1,NMAT)

2*NORD*NMAT=NUMBER OF WORDS

NGPB(L,J)  NUMBER OF SOURCE GROUPS THAT CAN SCATTER INTO GROUP N-

IFSG(L,J)  GROUP NUMBER OF THE FIRST SOURCE GROUP

L       LEGENDRE ORDER NUMBER

J       MATERIAL NUMBER

---------------------------------------------------------------------

SCATTERING SUB-BLOCK FOR GROUP N

PRESENT IF NORD.GT.0

(SCAT(I),I=1,NTAB)

NTAB*MULT=NUMBER OF WORDS

SCAT(I) SCATTERING CROSS SECTION

NTAB  TABLE LENGTH OF THE CROSS SECTIONS FOR SCATTERING

INTO GROUP N. THIS IS FOR ALL MATERIALS AND ALL

LEGENDRE ORDERS, THUS IT IS THE SUM OF NGPB(L,J)

OVER L FROM 1 TO NORD AND OVER J FROM 1 TO NMAT.

THE SCATTERING CROSS SECTIONS ARE PACKED IN BANDS,

ONE FOR EACH LEGENDRE ORDER AND MATERIAL. EACH BAND-

CONTAINS THE NGPB GROUPS WHICH SCATTER INTO GROUP

N. THE FIRST SOURCE GROUP NUMBER IS IFSG AND

THE LAST IS IFSG-NGPB+1. THE NORD BANDS FOR THE

FIRST MATERIAL APPEAR FIRST (P0, P1, ...) FOLLOWED-

BY THE NORD BANDS FOR THE SECOND, ETC.

HIGHER LEGENDRE ORDER SCATTERING CROSS SECTIONS

INCLUDE A 2*L+1 FACTOR WHERE L IS THE LEGENDRE

ORDER.

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.

FILE DESCRIPTIONS Version 3.0 15-31
CODE DEPENDENT INTERFACE FILES

AMFLUX

---

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
---

---

CR
ADJOINT MOMENTS FLUXES ON MULTIDIMENSIONAL INTERVALS
(2D RECORD)
---

CL
((FLUX(M,I),M=1,NORD),I=1,NINTI) ----NOTE STRUCTURE BELOW---
---

CW
NORD*NINTI=NUMBER OF WORDS
---

CD
DO 1 K=1,NINTK
---

CD
DO 1 J=1,NINTJ
---

CD
1 READ (N) *LIST AS ABOVE*
---

CD
FLUX(M,I)
ADJOINT FLUX MOMENTS ON FIRST DIMENSION
---

CD
INTERVALS.
---

---

CEO
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.

---

FILE STRUCTURE

**RECORD TYPE**
- PRESENT IF

**FILE IDENTIFICATION**
- ALWAYS

**FILE CONTROL**
- ALWAYS

**COMPATABILITY CODE**
- ALWAYS

**MATERIAL NAMES**
- ALWAYS

**ZONE NAMES**
- ALWAYS

**NUMBER OF MATERIALS PER ZONE**
- MPZTOT.NE.0

**MATERIAL LIST FOR ALL ZONES**
- MPZTOT.NE.0

**MATERIAL CONCENTRATIONS**
- MPZTOT.NE.0

**MATERIAL CONCENTRATION FACTORS**
- MPZTOT.NE.0

**CONCENTRATION MODIFIER**
- MPZTOT.NE.0

---

FILE IDENTIFICATION

**HNAME**
- HUSE(I), I=1,2, IVERS

**1+3*MULT-NUMBER OF WORDS**

**HNAME**
- HOLLERITH FILE NAME - ASGMAT - (A6)

**HUSE(I)**
- HOLLERITH USER IDENTIFICATION (A6)

**IVERS**
- FILE VERSION NUMBER

**MULT**
- DOUBLE PRECISION PARAMETER
  - 1- A6 WORD IS SINGLE WORD
  - 2- A6 WORD IS DOUBLE PRECISION WORD

---

FILE CONTROL

**MT, NZONE, MPZTOT**

**3-NUMBER OF WORDS**

**1 MT**
- NUMBER OF MATERIALS

**2 NZONE**
- NUMBER OF ZONES

**3 MPZTOT**
- IN-SOLVER MIXING TABLE LENGTH

**4 FMMIX**
- 0/1 NO/YES VOL FRACTION MIXING BY FINE MESH
CODE DEPENDENT INTERFACE FILES

ASGMAT

-------------------------------------------------------------------
CR COMPATABILITY CODE WORDS
CL CODE1, CODE2
CW 2*MULT=NUMBER OF WORDS
CD 1 CODE1 DATE OF THE MACRXS FILE TO WHICH ASGMAT APPLIES
CD 2 CODE2 TIME OF THE MACRXS FILE TO WHICH ASGMAT APPLIES

-------------------------------------------------------------------
CR MATERIAL NAMES
CL (MATNAM(I),I=1,MT)
CW MT*MULT=NUMBER OF WORDS
CD MATNAM(I) HOLLERITH NAME FOR THE I-TH MATERIAL

-------------------------------------------------------------------
CR ZONE NAMES
CL (ZONNAM(I),I=1,NZONE)
CW NZONE*MULT=NUMBER OF WORDS
CD ZONNAM(I) HOLLERITH NAME FOR THE I-TH ZONE

-------------------------------------------------------------------
CR NUMBER OF MATERIALS PER ZONE
CC PRESENT IF MPZTOT.NE.0
CL (NUMZON(I),I=1,NZONE)
CW NZONE=NUMBER OF WORDS
CD NUMZON(I) NUMBER OF MATERIALS IN THE I-TH ZONE

-------------------------------------------------------------------
CR MATERIAL LIST FOR ALL ZONES
CC PRESENT IF MPZTOT.NE.0
CL (MATLST(I),I=1,MPZTOT)
CW MPZTOT=NUMBER OF WORDS
CD MATLST PACKED LIST OF MATERIAL NUMBERS. MATERIALS FOR ZONE 1-
CD FOLLOWED BY THOSE FOR ZONE 2, THEN ZONE3, ETC.
CODE DEPENDENT INTERFACE FILES

MATERIAL CONCENTRATIONS

PRESENT IF MPZTOT.NE.0

(CONC(I), I=1, MPZTOT)

MPZTOT*MULT=NUMBER OF WORDS

CONC(I) CONCENTRATION OF THE I-TH MATERIAL IN THE MATLST ARRAY

MATERIAL CONCENTRATION FACTORS

PRESENT IF MPZTOT.NE.0

(C1(I), I=1, MPZTOT)

MPZTOT*MULT=NUMBER OF WORDS

C1(I) CONCENTRATION FACTOR FOR THE I-TH MATERIAL IN THE MATLST ARRAY

CONCENTRATION MODIFIER

PRESENT IF MPZTOT.NE.0

CMOD

1*MULT=NUMBER OF WORDS

CMOD INPUT VALUE OF THE CONCENTRATION MODIFIER

FILE DESCRIPTIONS Version 3.0

15:35
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.

```
DATE 01/28/95

CF AZMFLX-IV
CE ADJOINT ZONE AVERAGED FLUX MOMENTS BY GROUP

CN ORDER OF GROUPS IS ACCORDING TO INCREASING ENERGY

CS FILE STRUCTURE

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 *************

CR FILE IDENTIFICATION

CL HNAME, (HUSE(I),I=1,2), IVERS
CW 1+3*MULT=NUMBER OF WORDS

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

CR SPECIFICATIONS (1D RECORD)

CL NDIM,NGROUP, NZONE, DUM, DUM, NORD, EFFK, POWER, OITNO
CW 9=NUMBER OF WORDS

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
```
CODE DEPENDENT INTERFACE FILES

| 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 |

-----------------------------------------------------------------------

CRADJOINT FLUX MOMENTS AVERAGED OVER EACH ZONE  
(2D RECORD)  

CL ((FLUX(M,I),M=1,NORD),I=1,NZONE)  

CW NORD*NZONE=NUMBER OF WORDS  

CD FLUX(M,I)  
ADJOINT FLUX MOMENT AVERAGES FOR EACH ZONE.  

CEOF
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.

FILE STRUCTURE

FILE IDENTIFICATION

FILE CONTROL

FILE DATA

RECORD TYPE

PRESENT IF

FILE IDENTIFICATION

ALWAYS

FILE CONTROL

ALWAYS

FILE DATA

ALWAYS

****** (REPEAT FOR ALL ISOTOPES)

****** (REPEAT FOR ALL LEGENDRE ORDERS)

******

ISOTOPE LABEL/ATOMIC WEIGHT PAIR NISOAW GT 0

FILE IDENTIFICATION

HNAME, (HUSE(I), I=1,2), IVERS

1+3*MULT=NUMBER OF WORDS

HNAME HOLLERITH FILE NAME - BXSLIB - (A6)

HUSE(I) HOLLERITH USER IDENTIFICATION (A6)

IVERS FILE VERSION NUMBER

MULT DOUBLE PRECISION PARAMETER

1- A6 WORD IS SINGLE WORD

2- A6 WORD IS DOUBLE PRECISION WORD

FILE CONTROL

NGROUP, NISO, IHM, IHT, IHS, MAXT, NISOAW, NXSREC, I2LP1

9=NUMBER OF WORDS
CODE DEPENDENT INTERFACE FILES

CD NGROUP NUMBER OF ENERGY GROUPS IN FILE

CD NISO NUMBER OF ISOTOPES IN FILE

CD IHM TABLE LENGTH (NUMBER OF CROSS SECTIONS FOR ONE ISOTOPE, FOR ONE GROUP, AND FOR ONE LEGENDRE 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 2L+1 FACTOR

CR FILE DATA

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)

CW (NISO+12+IHT-3)*MULT+(3*NGROUP+1)*MULT+NISO=NUMBER OF WORDS

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 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

CR CROSS SECTION SET FOR ISOTOPE I AND LEGENDRE ORDER M

CL ((C(I,J), I=1,IHM), J=1,NGROUP)

CW IHM*NGROUP*MULT=NUMBER OF WORDS

CR HOLLERITH ISOTOPE LABEL/ATOMIC WEIGHT PAIR SET

CC PRESENT IF NISOAW.GT.0

CL (ATWTP(I), I=1,2*NISOAW)

CW 2*NISOAW*MULT=NUMBER OF WORDS

CEOFILE
The EDITIT code-dependent interface file contains information specific to the Edit Module, mainly information from Block VI of the card-image input.

FILE STRUCTURE

 RECORD TYPE PRESENT IF
 FILE IDENTIFICATION ALWAYS
 RAW CONTROLS AND DIMENSIONS ALWAYS
 DEFAULT CONTROLS AND DIMENSIONS ALWAYS
 RAW FLOATING INPUT DATA ALWAYS
 FINE GROUPS PER BROAD GROUP ALWAYS
 ZONE NUMBERS NZNS.NE.0
 POINT NUMBERS NIPE.NE.0
 IPLANE NUMBERS NIPLNE.NE.0
 JPLANE NUMBERS NJPLNE.NE.0
 KPLANE NUMBERS NKPLNE.NE.0
 CROSS SECTION POSITIONS NPOS.NE.0
 ISO TOPE NUMBERS TO EDIT NISO.NE.0
 MATERIAL NUMBERS TO EDIT MACRO.NE.0
 CONSTITUENT NUMBERS TO EDIT NCONS.NE.0
 RESPONSE FUNCTION NAMES IDOSE.NE.0
 ********** (REPEAT FOR ALL RESPONSE FUNCTIONS)
 * RESPONSE FUNCTION ENERGY VECTOR IDOSE.NE.0
 * RESPONSE FUNCTION X SPATIAL VECTOR IDOSE.NE.0
 * RESPONSE FUNCTION Y SPATIAL VECTOR IDOSE.NE.0.AND.IDIMEN.GT.1
 * RESPONSE FUNCTION Z SPATIAL VECTOR IDOSE.NE.0.AND.IDIMEN.GT.2
 **********
 CROSS SECTION SUMMING ARRAY - IXSUM.NE.0
 RESPONSE FUNCTION SUMMING ARRAY IRSUM.NE.0
 FINE X MESH DENSITY FACTORS IDEN.NE.0
 FINE Y MESH DENSITY FACTORS IDEN.NE.0.AND.IDIMEN.GT.1
 FINE Z MESH DENSITY FACTORS IDEN.NE.0.AND.IDIMEN.GT.2

FILE IDENTIFICATION

 HNAME, (HUSE(I),I=1,2), IVERS
 1+3*MULT=NUMBER OF WORDS
 HNAME HOLLERITH FILE NAME - EDITIT - (A6)
 HUSE(I) HOLLERITH USER IDENTIFICATION (A6)
CODE DEPENDENT INTERFACE FILES

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
RAW CONTROLS AND DIMENSIONS
C
CR  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 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 CCCCRZFLUX 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
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
FILE DESCRIPTIONS  Version 3.0  15-41
CODE DEPENDENT INTERFACE FILES

CW  2*MULT-NUMBER OF WORDS
CD  1  POWER  NORMALIZE TO POWER
CD  2  MEVPER  MEV RELEASED PER FISSION

CR  FINE GROUPS PER BROAD GROUP
CL  (ICOLL(G),G=1,NBG)
CW  NBG=NUMBER OF WORDS
CD  Icoll(G) NUMBER OF FINE GROUPS IN BROAD GROUP G

CR  ZONE NUMBERS
CC  PRESENT IF NZNS.GT.0
CL  (EDZONE(I),I=1,IT*JT*KT)
CW  IT*JT*KT=NUMBER OF WORDS
CD  EDZONE(I) EDIT ZONE NUMBER FOR THE I-TH FINE MESH

CR  POINTS TO EDIT
CC  PRESENT IF NIPE.GT.0
CL  (POINTS(I),I=1,NIPE)
CW  NIPE=NUMBER OF WORDS
CD  POINTS(I) NUMBER OF THE I-TH POINT TO EDIT

CR  IPLANES TO EDIT
CC  PRESENT IF NIPLNE.GT.0
CL  (IPLNES(I),I=1,NIPLNE)
CW  NIPLNE=NUMBER OF WORDS
CD  IPLNES(I) NUMBER OF THE I-TH PLANE TO EDIT

CR  JPLANES TO EDIT
CC  PRESENT IF NJPLNE.GT.0
CL  (JPLNES(I),I=1,NJPLNE)
CW  NJPLNE=NUMBER OF WORDS

15-42 FILE DESCRIPTIONS Version 3.0
CODE DEPENDENT INTERFACE FILES

CD     JPLNES(I) NUMBER OF THE J-TH PLANE TO EDIT
C
C---------------------------------------------------------
C
C     KPLANES TO EDIT
CR     KPLANES TO EDIT
CC     PRESENT IF NKPLNE.GT.0
CL     (KPLNES(I),I=1,NKPLNE)
CW     NKPLNE=NUMBER OF WORDS
CD     KPLNES(I) NUMBER OF THE K-TH PLANE TO EDIT
C
C---------------------------------------------------------
C
CR     CROSS SECTION POSITIONS TO EDIT
CC     PRESENT IF NPOS.GT.0
CL     (EDXS(I),I=1,NPOS)
CW     NPOS=NUMBER OF WORDS
CD     EDXS(I) POSITION NUMBER TO EDIT(IN NUMERIC FORM)
C
C---------------------------------------------------------
C
CR     ISOTOPE NUMBERS TO EDIT
CC     PRESENT IF NISO.GT.0
CL     (EDISOS(I),I=1,NISO)
CW     NISO=NUMBER OF WORDS
CD     EDISOS(I) ISOTOPE NUMBER TO EDIT(IN NUMERIC FORM)
C
C---------------------------------------------------------
C
CR     MATERIAL NUMBERS TO EDIT
CC     PRESENT IF MACRO.GT.0
CL     (EDMATS(I),I=1,MACRO)
CW     MACRO=NUMBER OF WORDS
CD     EDMATS(I) MATERIAL NUMBER TO EDIT(IN NUMERIC FORM)
C
C---------------------------------------------------------
C
CR     CONSTITUENT NUMBERS TO EDIT
CC     PRESENT IF NCONS.GT.0
CL     (EDCONS(I),I=1,NCONS)
CW     NCONS=NUMBER OF WORDS
CODE DEPENDENT INTERFACE FILES

EDCONS(I) CONSTITUENT NUMBER TO EDIT (IN NUMERIC FORM)

RESPONSE FUNCTION NAMES
PRESENT IF IDOSE.GT.0
(RSFNAM(I), I=1, IDOSE)
IDOSE*MULT=NUMBER OF WORDS
RSFNAM(I) HOLLERITH NAME FOR THE I-TH RESPONSE FUNCTION

RESPONSE FUNCTION ENERGY VECTOR
PRESENT IF IDOSE.GT.0
(RSFE(I), I=1, NGROUP)
NGROUP*MULT=NUMBER OF WORDS
RSFE(I) RESPONSE FOR GROUP I

RESPONSE FUNCTION SPATIAL VECTOR IN X
PRESENT IF IDOSE.LT.0
(RSFX(I), I=1, IT)
IT*MULT=NUMBER OF WORDS
RSFX(I) RESPONSE FUNCTION FOR FINE MESH I

RESPONSE FUNCTION SPATIAL VECTOR IN Y
PRESENT IF IDOSE.LT.0 .AND. IDIMEN.GT.1
(RSFY(J), J=1, JT)
JT*MULT=NUMBER OF WORDS
RSFY(J) RESPONSE FUNCTION FOR FINE MESH J

RESPONSE FUNCTION SPATIAL VECTOR IN Z
PRESENT IF IDOSE.LT.0 .AND. IDIMEN.GT.2
(RSFZ(K), K=1, KT)
KT*MULT=NUMBER OF WORDS
C    RSFZ(K)  RESPONSE FUNCTION FOR FINE MESH K
C
C---------------------------------------------------------------------
C    CROSS SECTION SUMMING ARRAY
C
CC    PRESENT IF IXSUM.GT.0
CL (MICSUM(I),I=1,IXSUM)
CW IXSUM=NUMBER OF WORDS
CD MICSUM INPUT SUMMING ARRAY IN NUMERIC FORM
C
C---------------------------------------------------------------------
C    RESPONSE FUNCTION SUMMING ARRAY
C
CC    PRESENT IF IRSUM.GT.0
CL (IRSUMS(I),I=1,IRSUM)
CW IRSUM=NUMBER OF WORDS
CD IRSUMS INPUT SUMMING ARRAY IN NUMERIC FORM
C
C---------------------------------------------------------------------
C    FINE MESH DENSITY VECTOR IN X
C
CC    PRESENT IF IDEN.GT.0
CL (XDF(I),I=1,IT)
CW IT*MULT=NUMBER OF WORDS
CD XDF(I) DENSITY FACTOR FOR THE I-TH FINE MESH
C
C---------------------------------------------------------------------
C    FINE MESH DENSITY VECTOR IN Y
C
CC    PRESENT IF IDEN.GT.0 .AND. IDIMEN.GT.1
CL (YDF(J),J=1,JT)
CW JT*MULT=NUMBER OF WORDS
CD YDF(J) DENSITY FACTOR FOR THE J-TH FINE MESH
C
C---------------------------------------------------------------------
C    FINE MESH DENSITY VECTOR IN Z
C
CC    PRESENT IF IDEN.GT.0 .AND. IDIMEN.GT.2
CL (ZDF(K),K=1,KT)
CW KT*MULT=NUMBER OF WORDS
C
C---------------------------------------------------------------------

FILE DESCRIPTIONS  Version 3.0
C
CD ZDF(K) DENSITY FACTOR FOR THE K-TH FINE MESH
C
CEOF
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,

\[
\sum_{g} (v \Sigma f_{g}^{i}) \phi_{g}^{i},
\]

The FISSRC file is automatically produced by the Solver Module whenever fissions are present.
CODE DEPENDENT INTERFACE FILES

FISSRC

C------------------------------------------------------------------------
CR SPECIFICATIONS (1D RECORD)
CL
CW 9 =NUMBER OF WORDS
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 WRITTEN
CD EFFK EFFECTIVE MULTIPLICATION FACTOR
CD POWER POWER IN WATTS TO WHICH FISSION IS NORMALIZED
CD NBLOK SET TO 1
C------------------------------------------------------------------------

C------------------------------------------------------------------------
CR FISSION SOURCE (2D RECORD)
CL
CW NINTI*NINTJ*MULT=NUMBER OF WORDS
CD DO 1 K=1, NINTK
CD 1 READ (N) *LIST AS ABOVE*
CD FISS(I,J) FISSION SOURCE WITHOUT VOLUME AT FINE MESH POINT (I,J) IN PLANE K, I.E., NUSIGF*FLUX
C------------------------------------------------------------------------

15-48

FILE DESCRIPTIONS Version 3.0
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 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-MESSES) 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)
```

FILE DESCRIPTIONS Version 3.0 15-49
FILE IDENTIFICATION (0V RECORD)

HNAME, (HUSE(I), I=1,2), IVERS

1+3*MULT

HNAME HOLLERITH FILE NAME - GEODST - (A6)
HUSE HOLLERITH USER IDENTIFICATION (A6)
IVERS FILE VERSION NUMBER
MULT DOUBLE PRECISION PARAMETER
1- A6 WORD IS SINGLE WORD
2- A6 WORD IS DOUBLE PRECISION WORD

FILE SPECIFICATIONS (1D RECORD)

IGOM, NZONE, NREG, NZCL, NCINTI, NCINTJ, NCINTK, NINTI, NINTJ, NINTK, IMB1, IMB2, JMB1, JMB2, KMB1, KMB2, NBS, NBCS, NIBCS, NZWBB, NTRIAAG, NRAASS, NTHPT, (NGOP(I), I=1,4)

IGOM GEOMETRY
0- POINT (FUNDAMENTAL MODE)
1- SLAB
2- CYLINDER
3- SPHERE
6- X-Y
7- R-Z
8- THETA-R
9- UNIFORM TRIANGULAR
10- HEXAGONAL (1 MESH POINT IN EACH HEXAGONAL ELEMENT)
11- R-THETA
12- R-THETA-Z
13- R-THETA-ALPHA
14- X-Y-Z
15- THETA-R-Z
16- THETA-R-ALPHA
17- UNIFORM TRIANGULAR-Z
18- HEXAGON-Z (MESH POINTS AS IN 10 ABOVE)
106- X-Y GENERALIZED
107- R-Z GENERALIZED

NZONE NUMBER OF ZONES (EACH HOMOGENEOUS IN NEUTRONICS PROBLEM - A ZONE CONTAINS ONE OR MORE REGIONS)
NREG NUMBER OF REGIONS
NZCL NUMBER OF ZONE CLASSIFICATIONS (EDIT PURPOSES)
NCINTI NUMBER OF FIRST DIMENSION COARSE MESH INTERVALS
NCINTJ NUMBER OF SECOND DIMENSION COARSE MESH INTERVALS. NCINTJ.EQ.1 FOR ONE DIMENSIONAL CASE.
NCINTK NUMBER OF THIRD DIMENSION COARSE MESH INTERVALS. NCINTK.EQ.1 FOR ONE AND TWO DIMENSIONAL CASES.
NINTI NUMBER OF FIRST DIMENSION FINE MESH INTERVALS
NINTJ NUMBER OF SECOND DIMENSION FINE MESH INTERVALS. NINTJ.EQ.1 FOR ONE DIMENSIONAL CASE.
NINTK  NUMBER OF THIRD DIMENSION FINE MESH INTERVALS
NINTK.EQ.1 FOR ONE AND TWO DIMENSION CASES.

IMB1  FIRST BOUNDARY ON FIRST DIMENSION
0 - ZERO FLUX (DIFFUSION)
1 - REFLECTED
2 - EXTRAPOLATED (DIFFUSION - DEL PHI/PHI
   = -C/D WHERE C IS GIVEN AS BNDC BELOW
   AND D IS THE GROUP DIFFUSION CONSTANT,
   TRANSFER - NO RETURN).
3 - REPEATING (PERIODIC) WITH OPPOSITE FACE
4 - REPEATING (PERIODIC) WITH NEXT ADJACENT
   FACE.
5 - INVERTED REPEATING ALONG THIS FACE.
6 - ISOTROPIC RETURN (TRANSFR)

NOTE FOR REPEATING CONDITIONS (3,4,5) - LET I1 DENOTE FIRST-
BOUNDARY ON FIRST DIMENSION, I2 THE SECOND BOUNDARY ON THE
FIRST DIMENSION, J1 THE FIRST BOUNDARY ON THE SECOND
DIMENSION, ETC. THEN THESE REPEATING BOUNDARY CONDITIONS
ONLY APPLY TO BOUNDARIES I1,I2,J1, AND J2. GOING IN ORDER
OF I1,J1,I2,J2, THE FIRST BOUNDARY WHICH IS INVOLVED
CARRIES THE DESIGNATOR DEFINING THE REPEATING CONDITION.

IMB2  LAST BOUNDARY ON FIRST DIMENSION
JMB1  FIRST BOUNDARY ON SECOND DIMENSION
JMB2  LAST BOUNDARY ON SECOND DIMENSION
KMB1  FIRST BOUNDARY ON THIRD DIMENSION
KMB2  LAST BOUNDARY ON THIRD DIMENSION
NBSS  NUMBER OF BUCKLING SPECIFICATIONS
0 - NONE
1 - SINGLE VALUE APPLIES EVERYWHERE
.NEQ.NZONE- ZONE DEPENDENT
M*NZONE - DATA IS GIVEN OVER ALL ZONES FOR
THE FIRST ENERGY GROUP, THEN FOR THE
NEXT GROUP, TO END OF LIST. IF
M.LT.NGROUP THEN THE M-TH GROUP DATA
APPLIES TO ALL ADDITIONAL GROUPS.
(2.LE.M.LE.NGROUP)

NBSS  NUMBER OF BUCKLING SPECIFICATIONS
0 - NONE
1 - SINGLE VALUE USED EVERYWHERE
6 - INDIVUAL VALUE GIVEN FOR EACH
EXTERNAL BOUNDARY. THE ORDERING OF THE
VALUES IS THE SAME AS THE ORDERING OF
THE BOUNDARY CONDITIONS.
6*M - SIX VALUES GIVEN FOR FIRST ENERGY
GROUP (ORDERED AS DESCRIBED ABOVE),
THEN 6 FOR THE NEXT GROUP, TO END OF
LIST. (2.LE.M.LE.NGROUP).
IF M.LT.NGROUP THEN THE M-TH GROUP DATA
APPLIES TO ALL REMAINING GROUPS.

NIBCS  NUMBER OF CONSTANTS FOR INTERNAL BOUNDARIES
0 - NONE
1 - SINGLE VALUE USED EVERYWHERE
.GT.1 - VALUES ARE GIVEN BY ENERGY GROUP
WITH NON-BLACK CONDITION INDICATED BY
ZERO ENTRY - LAST VALUE APPLIES TO
ADDITIONAL GROUPS

NZWBB  NUMBER OF ZONES WHICH ARE BLACK ABSORBERS
NTRIAG  TRIANGULAR/HEXAGONAL GEOMETRY OPTION
0 - REGION OF SOLUTION IS A RHOMBUS IN
WHICH THE 1ST AND 2ND DIMENSION AXES
INTERSECT AT AN ANGLE OF 120 DEGREES.
1 - REGION OF SOLUTION IS A RHOMBUS IN WHICH THE 1ST AND 2ND DIMENSION AXES INTERSECT AT AN ANGLE OF 60 DEGREES.

2 - REGION OF SOLUTION IS A RECTANGLE. THE BOUNDARIES 11 AND 12 BISECT MESH TRIANGLES. SEE NTHPT BELOW. (IGOM=9,17 ONLY)

3 - REGION OF SOLUTION IS AN EQUILATERAL, 60 DEGREE TRIANGLE. (IGOM=9,17 ONLY)

4 - REGION OF SOLUTION IS A 30-60 DEGREE RIGHT TRIANGLE IN WHICH THE 1ST AND 2ND DIMENSION AXES INTERSECT AT THE 30 DEGREE ANGLE. (IGOM=9,17 ONLY)

5 - REGION OF SOLUTION IS A RHOMBUS IN WHICH THE 1ST AND 2ND DIMENSION AXES INTERSECT AT AN ANGLE OF 30 DEGREES. (IGOM=9,17 ONLY)

NRASS REGION ASSIGNMENTS
0 - TO COARSE MESH
1 - TO FINE MESH
2 - TO SUBMESH

NTHPT ORIENTATION OF FIRST FINE MESH INTERVAL IN TRIANGULAR GEOMETRIES. NTRIAG=2 ONLY.
1- TRIANGLE(1,1) POINTS AWAY FROM FIRST DIMENSION AXIS, I.E., NO INTERNAL MESH LINE INTERSECTS THE ORIGIN.
2- TRIANGLE(1,1) POINTS TOWARD THE FIRST DIMENSION AXIS, I.E., AN INTERNAL MESH LINE INTERSECTS THE ORIGIN.

NGOP RESERVED

ONE DIMENSIONAL COARSE MESH INTERVAL BOUNDARIES AND FINE MESH INTERVALS (2D RECORD)
PRESENT IF IGOM.GT.0 AND IGOM.LE.3
(XMESH(I), I=1,NCBNDI), (IFINTS(I), I=1,NCINTI)
NCBNDI*MULT+NCINTI
XMESH COARSE MESH BOUNDARIES, FIRST DIMENSION IFINTS NUMBER OF EQUALLY SPACED FINE MESH INTERVALS PER COARSE MESH INTERVAL, FIRST DIMENSION.
NCBNDI NCINTI+1, NUMBER OF FIRST DIMENSION COARSE MESH BOUNDARIES UNITS ARE CM FOR LINEAR DIMENSIONS AND RADIANS FOR ANGULAR DIMENSIONS

TWO DIMENSIONAL COARSE MESH INTERVAL BOUNDARIES AND FINE MESH INTERVALS (3D RECORD)
PRESENT IF IGOM.GE.6 AND IGOM.LE.11
(XMESH(I), I=1,NCBNDI), (YMESH(J), J=1,NCBNDJ),
1(IFINTS(I), I=1,NCINTI), (JFINTS(J), J=1,NCINTJ)
CODE DEPENDENT INTERFACE FILES GEO DST

(CW (NCBNDI+NCBNDJ) *MULT+NCINTI+NCINTJ

CD YME SH COARSE MESH BOUNDARIES, SECOND DIMENSION
CD JFINTS NUMBER OF EQUALLY SPACED FINE MESH INTERVALS PER COARSE MESH INTERVAL, SECOND DIMENSION.
CD NCBNDJ NCINTJ+1, NUMBER OF SECOND DIMENSION COARSE MESH BOUNDARIES

CC FOR UNIFORM-TRIANGULAR-MESH GEOMETRY (IGOM = 9) THE LENGTH (L) OF THE SIDE OF A MESH TRIANGLE MUST BE GIVEN BY THE EXPRESSION

L = 2.*(XMSH(2)-XMSH(1))/IFINTS(1)

CC FOR UNIFORM-HEXAGONAL-MESH GEOMETRY (IGOM = 10) THE FLAT-TO-FLAT DISTANCE (FTF) ACROSS A MESH HEXAGON MUST BE GIVEN BY THE EXPRESSION

FTF = (XMSH(2)-XMSH(1))/IFINTS(1)

CC----------------------------------------------------------------------

(CW 3D MESH INTERVAL BOUNDARIES AND FINE MESH INTERVALS (4D RECORD)

CC PRESENT IF IGOM.GE.12 .AND. IGOM.LT.100

CL (XMSH(I),I=1,NCBNDI),(YMSH(J),J=1,NCBNDJ),
CL 1(ZMSH(K),K=1,NCBNDK),(IFINTS(I),I=1,NCINTI),
CL 2(IFINTS(J),J=1,NCINTJ),(KFINTS(K),K=1,NCINTK)

(CW (NCBNDI+NCBNDJ+NCBNDK) *MULT+NCINTI+NCINTJ+NCINTK

CD ZMSH COARSE MESH BOUNDARIES, THIRD DIMENSION
CD KFINTS NUMBER OF EQUALLY SPACED FINE MESH INTERVALS PER COARSE MESH INTERVAL, THIRD DIMENSION.
CD NCBNDK NCINTK+1, NUMBER OF THIRD DIMENSION COARSE MESH BOUNDARIES

CC FOR UNIFORM-TRIANGULAR-MESH GEOMETRY (IGOM = 17) THE LENGTH (L) OF THE SIDE OF A MESH TRIANGLE MUST BE GIVEN BY THE EXPRESSION

L = 2.*(XMSH(2)-XMSH(1))/IFINTS(1)

CC FOR UNIFORM-HEXAGONAL-MESH GEOMETRY (IGOM = 18) THE FLAT-TO-FLAT DISTANCE (FTF) ACROSS A MESH HEXAGON MUST BE GIVEN BY THE EXPRESSION

FTF = (XMSH(2)-XMSH(1))/IFINTS(1)

CC----------------------------------------------------------------------

(CW GEOMETRY DATA (5D RECORD)

CC PRESENT IF IGOM.GT.0 OR NBS.GT.0

CL (VOLR(N),N=1,NREG), (BSQ(N),N=1,NBS), (BNDC(N),N=1,NBCS),
CL (BNCl(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 VOLR REGION VOLUMES (CC)
C BSQ BUCKLING (B**2) VALUES (CM**-2)
C BNDC BOUNDARY CONSTANTS (DEL PHI/PHI =-C/D)
C BNCl INTERNAL BLACK BOUNDARY CONSTANTS
C NZHBB ZONE NUMBERS WITH BLACK ABSORBER CONDITIONS
C NZC ZONE CLASSIFICATIONS

FILE DESCRIPTIONS Version 3.0 15-53
CODE DEPENDENT INTERFACE FILES

CD NZNR ZONE NUMBER ASSIGNED TO EACH REGION

CR REGION ASSIGNMENTS TO COARSE MESH INTERVALS (6D RECORD)
CC PRESENT IF IGOM.GT.0 AND NRASS.EQ.0
CL ((MR(I,J), I=1,NCINTI),J=1,NCINTJ) ----NOTE STRUCTURE BELOW----
CW NCINTI*NCINTJ
CS DO 1 K=1,NCINTK
CS 1 READ(N) *LIST AS ABOVE*
CD MR REGION NUMBERS ASSIGNED TO COARSE MESH INTERVALS

CR REGION ASSIGNMENTS TO FINE MESH INTERVALS (7D RECORD)
CC PRESENT IF IGOM.GT.0 AND NRASS.EQ.1
CL ((MR(I,J), I=1,NINTI),J=1,NINTJ) ----NOTE STRUCTURE BELOW----
CW NINTI*NINTJ
CS DO 1 K=1,NINTK
CS 1 READ(N) *LIST AS ABOVE*
CD MR REGION NUMBERS ASSIGNED TO FINE MESH INTERVALS

THE FULL PROBLEM AREA MAY BE DECOMPOSED INTO A SET OF SEPARATE SMALLER AREAS, EACH WITH A MESH OF ITS OWN, CALLED A SUBMESH. EACH SUB-MESH IS THEN DESCRIBED SEPARATELY. SOME VERTICES ON THE EXTERIOR OF A SUBMESH MAY ALSO LIE IN A CONTIGUOUS SUBMESH. NOTICE THERE IS NO IMPLIED CONSTRAINT THAT EACH EXTERIOR SUBMESH VERTEX MATCH A VERTEX ON AN ADJACENT SUBMESH. THUS, ONE SUBMESH MAY BE FINER THAN ITS ADJOINING ONE.

CR GENERALIZED 2D SPECIFICATIONS (8D RECORD)
CC PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107
CL NSMESH,NZONIG,NOBJS,NCOMPS,NEB,
CL MAXVTX,MAXICN,MAXINT,NUMVTX,NUMICN,NUMINT,NUMCOM,NCOMGE,NAMLEN
CW 14
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 NCOMGE | TOTAL NUMBER OF SUBCOMPONENTS IN GEOMETRY |
| CD NAMLEN | NUMBER OF A6 WORDS IN A SUBMESH, OBJECT, OR COMPONENT NAME |

**SUBMESH JOINING DATA (9D RECORD)**

- PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

- CL XYEPS
- CW 1*MULT = NUMBER OF WORDS
- CD XYEPS EPSILON FOR POSITION IDENTITY

**NOTE:** WHEN JOINING TWO SUBMESHES TO FORM A LARGER MESH, SOME INTERFACE VERTEXES MAY BE DESCRIBED TWICE, ONCE IN EACH SUBMESH. IF THEY LIE WITHIN XYEPS OF EACH OTHER, THEY MAY BE CONSIDERED TO BE THE SAME VERTEX.

**SUB-MESH NAME (10D RECORD)**

- PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

- CL SNAME
- CW 1*MULT*NAMLEN = NUMBER OF WORDS
- CD SNAME HOLLERITH SUB-MESH NAME (A6)

**SUB-MESH SPECIFICATIONS (11D RECORD)**

- PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

- CL NVTX,ILR,IM,JM,NEDGES,NINT
- CW 6 = NUMBER OF WORDS
- CD NVTX NUMBER OF VERTEXES IN THIS SUB-MESH
- CD ILR FLAG FOR LOGICALLY RECTANGULAR SUB-MESH DOMAIN
  - 0 - NOT A LOGICALLY RECTANGULAR DOMAIN
  - 1 - LOGICALLY RECTANGULAR DOMAIN
- CD IM NUMBER OF VERTEXES IN THE X DIRECTION IF THE SUB-MESH HAS A LOGICALLY RECTANGULAR DOMAIN
- CD JM NUMBER OF VERTEXES IN THE Y DIRECTION IF THE 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

FILE DESCRIPTIONS  Version 3.0
CODE DEPENDENT INTERFACE FILES

CR SUB-MESH VERTEXES (12D RECORD)
CC PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107
CL (XVERT(I), I=1, NVTX), (YVERT(I), I=1, NVTX)
CW 2*NVTX*MULT = NUMBER OF WORDS
CD XMESH X COORDINATES OF VERTEXES
CD YMESH Y COORDINATES OF VERTEXES

FOR A LOGICALLY RECTANGULAR DOMAIN, THE EDGES AND
INTERVALS ARE AUTOMATICALLY DEFINED. THE FIRST IM+1
VERTEXES ABOVE FORM, IN ORDER, THE FIRST LINE IN THE FIRST
DIMENSION, THE SECOND IM+1 POINTS FORM THE SECOND LINE.
A LINE CONTAINS IM EDGES, EACH FORMED BY CONNECTING
SUCCESSIVE VERTEXES AND EACH VERTEX IN THE FIRST LINE IS
CONNECTED BY AN EDGE TO THE CORRESPONDING VERTEX IN THE
SECOND LINE. THIS SERIES OF EDGES ENCLOSES THE FIRST IM
INTERVALS. CONTINUING THIS PROCESS THRU ALL THE LINES
GENERATES THE FULL LOGICALLY RECTANGULAR MESH STRUCTURE.

CR SUB-MESH INTERVAL CORNER COUNTS (13D RECORD)
CC PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND ILR.EQ.0
CL (NICN(I), I=1, NINT)
CW NINT = NUMBER OF WORDS
CD NICN(I) NUMBER OF CORNERS IN INTERVAL I

CR SUB-MESH INTERVALS (14D RECORD)
CC PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND ILR.EQ.0
CL (ICN(I), I=1, NICNT)
CW NICNT = NUMBER OF WORDS
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.

CR ZONING NAME (15D RECORD)

FILE DESCRIPTIONS  Version 3.0
CODE DEPENDENT INTERFACE FILES

PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND NRASS.EQ.2

ZNAME

1*MULT*NAMLEN = NUMBER OF WORDS

ZNAME HOLLERITH ZONING NAME (A6)

-----------------------------------------------------------------------

ZONING SPECIFICATIONS (16D RECORD)

PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND NRASS.EQ.2

LENZON

1 = NUMBER OF WORDS

LENZON NUMBER OF INTERVALS IN THIS ZONING

-----------------------------------------------------------------------

REGION ASSIGNMENTS BY SUBMESH INTERVAL (17D RECORD)

PRESENT IF (IGOM.EQ.106 OR IGOM.EQ.107) AND NRASS.EQ.2

(MR(I), I=1, LENZON)

LENZON = NUMBER OF WORDS

MR REGION NUMBERS ASSIGNED TO MESH INTERVALS

-----------------------------------------------------------------------

OBJECT NAMES (18D RECORD)

PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

(OBNAME(I), I=1, NOBJS)

NOBJS*MULT*NAMLEN = NUMBER OF WORDS

OBNAME HOLLERITH OBJECT NAME (A6)

-----------------------------------------------------------------------

OBJECT SPECIFICATIONS (19D RECORD)

PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

(KSMESH(I), I=1, NOBJS), (KZONIG(I), I=1, NOBJS)

2*NOBJS = NUMBER OF WORDS

KSMESH INDEX OF SUB-MESH FOR THIS OBJECT

KZONIG INDEX OF ZONING FOR THIS OBJECT
CODE DEPENDENT INTERFACE FILES

---

COMPONENT NAME (20D RECORD)

PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

CMNAME

MULT*NAMLEN = NUMBER OF WORDS

CMNAME HOLLERITH COMPONENT NAME (A6)

---

COMPONENT SPECIFICATIONS (21D RECORD)

PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

NSCOMS

1 = NUMBER OF WORDS

NSCOMS NUMBER OF SUBCOMPONENTS IN COMPONENT

---

COMPONENT DESCRIPTION (22D RECORD)

PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

(X(I), I=1,NSCOMS), (Y(I), I=1,NSCOMS), (THETA(I), I=1,NSCOMS),

(XTRANS(I), I=1,NSCOMS), (YTRANS(I), I=1,NSCOMS),

(IOBJ(I), I=1,NSCOMS)

5*NSCOMS*MULT+NSCOMS = NUMBER OF WORDS

X0 Y COORDINATE OF ROTATION POINT

Y0 Y COORDINATE OF ROTATION POINT

THETA ROTATION ANGLE, CLOCKWISE DEGREES

XTRANS X COORDINATE TRANSLATION DISTANCE

YTRANS Y COORDINATE TRANSLATION DISTANCE

IOBJ INDEX OF AN OBJECT OR COMPONENT

POSITIVE FOR OBJECT

NEGATIVE FOR COMPONENT

---

GEOMETRY NAME (23D RECORD)

PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107

CMNAME

MULT*NAMLEN = NUMBER OF WORDS

GENAME HOLLERITH GEOMETRY NAME (A6)
CODE DEPENDENT INTERFACE FILES

CR GEOMETRY SPECIFICATIONS (24D RECORD)
CC PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107
CL NSCOMS
CW 1 = NUMBER OF WORDS
CD NSCOMS NUMBER OF SUBCOMPONENTS IN GEOMETRY

CR GEOMETRY DESCRIPTION (25D RECORD)
CC PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107
CL (XO(I), I=1,NSCOMS), (YO(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)
CW 5*NSCOMS*MULT+NSCOMS = NUMBER OF WORDS
CD XO X COORDINATE OF ROTATION POINT
CD YO 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
   POSITIVE FOR OBJECT
   NEGATIVE FOR COMPONENT

CR BOUNDARY SEGMENT, CONDITION (26D RECORD)
CC PRESENT IF IGOM.EQ.106 OR IGOM.EQ.107 AND NEB.GT.0
CL X1,Y1,X2,Y2,BTHETA,IB,IR
CW 5*MULT+2 = NUMBER OF WORDS
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
   RECIPIENT SEGMENT (IF ANY). UNITS ARE DEGREES
CD IB BOUNDARY CONDITION
   SAME OPTIONS AS IMB1 IN 1D RECORD, PLUS:
   7 - EXITING ANGULAR FLUXES REENTER ELSEWHERE
   8 - EXITING ANGULAR FLUXES ARE MIRROR IMAGED ABOUT THE PERPENDICULAR TO THE
   SEGMENT AND THEN REENTERED ELSEWHERE
CD IR INDEX OF THE BOUNDARY SEGMENT THAT IS THE
   RECIPIENT OF FLUXES EXITING THIS SEGMENT

FILE DESCRIPTIONS  Version 3.0  15-59
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.
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
CN THIS FILE PROVIDES THE FINE MESH MIXING PRESCRITIONS
C
C---------------------------------------------------------------------
C
CD NOTE THAT DOUBLE PRECISION VOLUME FRACTIONS ARE GIVEN WHEN MULT=2
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
CEOF
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
CN THIS FILE PROVIDES A BASIC BROAD GROUP LIBRARY, ORDERED BY GROUP
CN ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY
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
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
```
CODE DEPENDENT INTERFACE FILES

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 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

-----------------------------------------------------------------------
FILE DATA
-----------------------------------------------------------------------

(CMAT(I),I=1,NMAT), (HED(J), J=1, NEDT), (VEL(N), N=1, NGROUP),
1(EMAX(N), N=1, NGROUP), EMIN

(NMAT*NEDT+2*NGROUP+1)*MULT=NUMBER OF WORDS

HMAT(I)  HOLLERITH MATERIAL LABEL FOR MATERIAL I (A6)
HED(J)  HOLLERITH LABEL FOR J-TH CROSS SECTION POSITION (A6)
VEL(N)  MEAN NEUTRON VELOCITY IN GROUP N (CM/SEC)
EMAX(N)  MAXIMUM ENERGY BOUND OF GROUP N (EV)
EMIN  MINIMUM ENERGY BOUND OF SET (EV)
NEDT  NED+NPRIN

THE FOUR BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:

HED(1) = 3HCHI
HED(2) = 6HNUSIGF
HED(3) = 5HTOTAL
HED(4) = 3HABS

ALSO PRESENT WHEN NPRIN=5 IS:

HED(5) = 5HTRANS

-----------------------------------------------------------------------

PRINCIPAL CROSS SECTIONS FOR GROUP N

((C(I,J), I=1,NMAT), J=1, NEDT)

NMAT*NEDT*MULT=NUMBER OF WORDS

C(I,J)  PRINCIPAL CROSS SECTIONS

BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:

J=1 FISSION SPECTRUM
J=2 FISSION NU*FISSION CROSS SECTION
J=3 TOTAL CROSS SECTION
J=4 ABSORPTION CROSS SECTION

ALSO PRESENT WHEN NPRIN=5 IS:

J=5 TRANSPORT CROSS SECTION

-----------------------------------------------------------------------

FILE DESCRIPTIONS  Version 3.0  15-65
CODE DEPENDENT INTERFACE FILES

MACRXS

C-----------------------------------------------------------------------
C 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-----------------------------------------------------------------------
C 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

FILE DESCRIPTIONS Version 3.0
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 EDGES FOR TWODANT CODE
C
C**********************************************************************

CD ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY. NOTE THAT DOUBLE PRECISION FLUXES ARE GIVEN WHEN MULT=2
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

CR FILE IDENTIFICATION
C
CL HNAME, (HUSE(I), I=1,2), IVERS
CODE DEPENDENT INTERFACE FILES

RAFLXM for TWODANT

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 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 EFFK  EFFECTIVE MULTIPLICATION FACTOR
CD POWER  POWER IN WATTS TO WHICH FLUX IS NORMALIZED

C-------------------------------------------------------

C-----------------------------------------------------------------------

C 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 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
CD NBDRYI  NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH
CD BOUNDARIES)

C-----------------------------------------------------------------------

C VERTICAL EDGE ANGULAR FLUX

C

FILE DESCRIPTIONS  Version 3.0
CODE DEPENDENT INTERFACE FILES

RAFLXM for TWODANT

CL  ((VEDGE(I,J),I=1,NINTI),J=1,NBDRYJ)
CW  NINTI*NBDRYJ*MULT=NUMBER OF WORDS
CD  VEDGE(I,J)  VERTICAL-EDGE-BOUNDARY ANGULAR FLUX
CD  NBDRYJ  NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH BOUNDARIES)

CEOF

VERTICAL EDGE

HORIZONTAL EDGE
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 EDGES FOR THREEDANT CODE
C C**************************************************************************************-

C THIS FILE PROVIDES THE EDGE ANGULAR FLUX AS ORDERED BY THE CODE

C ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY. NOTE THAT DOUBLE PRECISION FLUXES ARE GIVEN WHEN MULT=2

C FILE STRUCTURE
C
C RECORD TYPE PRESENT IF
C =========== ..===. .===. .—-== ===. .===. =====—.======
C FILE IDENTIFICATION ALWAYS
C FILE CONTROL ALWAYS
C
C *********** (REPEAT FOR ALL GROUPS)
C (GROUP 1 IS FIRST)
C *********** (REPEAT FOR ALL FRONT-GOING DIRECTIONS)
C * COMPUTATIONAL COSINES ALWAYS
C * CELL BACK-EDGE ANGULAR FLUX FOR BACK PLANE
C * ********
C * ******** (REPEAT FOR ALL Z-INTERVALS)
C (INTERVAL NINTK IS FIRST)
C * ******** (REPEAT FOR ALL FRONT-GOING DIRECTIONS)
C * * COMPUTATIONAL COSINES ALWAYS
C * * HORIZONTAL-EDGE ANGULAR FLUX ALWAYS
C * * VERTICAL-EDGE ANGULAR FLUX ALWAYS
C * * CELL FRONT-EDGE ANGULAR FLUX ALWAYS
C * * ********
C * * ********
CODE DEPENDENT INTERFACE FILES

RAFLXM for THREE-DANT

CS * ============================= (REPEAT FOR ALL BACK-GOING DIRECTIONS)
CS * * COMPUTATIONAL COSINES ALWAYS
CS * * CELL FRONT-EDGE ANGULAR FLUX FOR FRONT PLANE
CS * * =============================
CS * * ============================= (REPEAT FOR ALL Z-INTERVALS)
CS * * (INTERVAL 1 IS FIRST)
CS * * * COMPUTATIONAL COSINES ALWAYS
CS * * * HORIZONTAL-EDGE ANGULAR FLUX ALWAYS
CS * * * VERTICAL-EDGE ANGULAR FLUX ALWAYS
CS * * * CELL BACK-EDGE ANGULAR FLUX ALWAYS
CS * * * =============================

CR FILE IDENTIFICATION

CL HNAME, (HUSE(I), I=1,2), IVERS

CW 1+3*MULT=NUMBER OF WORDS

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

CR SPECIFICATIONS (1D RECORD)

CL NDIM,NGROUP,NINTI,NINTJ,NINTK,EFFK,POWER

CW 8 =NUMBER OF WORDS

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 EFFK EFFECTIVE MULTIPLICATION FACTOR
CD POWER POWER IN WATTS TO WHICH FLUX IS NORMALIZED

CR COMPUTATIONAL COSINES

CL XMU,XETA,XI,WEIGHT,M,K

CW 6*MULT

CD XMU MU COSINE
CD XETA ETA COSINE

FILE DESCRIPTIONS Version 3.0 15-71
CODE DEPENDENT INTERFACE FILES

RAFLXM for THREEDANT

CD XI XI COSINE
CD WEIGHT WEIGHT
CD M ANGLE INDEX IN THE OCTANT
CD K Z-PLANE NUMBER

-------------------------------------------------------
CR CELL FRONT-EDGE ANGULAR FLUX

CL ((FBEDGE(I,J),I=NINTI),J=1,NINTJ)

Cw NINTI*NINTJ*MULT=NUMBER OF WORDS

CD FBEDGE(I,J) CELL FRONT-EDGE ANGULAR FLUX

-------------------------------------------------------
CR HORIZONTAL-EDGE ANGULAR FLUX

CL ((HEDGE(I,J),I=1,NBDRYI),J=1,NINTJ)

Cw NBDRYI*NINTJ*MULT=NUMBER OF WORDS

CD NBDRYI NINTI+1 (NUMBER OF FIRST DIMENSION FINE MESH BOUNDARIES)

-------------------------------------------------------
CR VERTICAL-EDGE ANGULAR FLUX

CL ((VEDGE(I,J),I=1,NINTI),J=1,NBDRYJ)

Cw NINTI*NBRDYJ*MULT=NUMBER OF WORDS

CD NBDRYJ NINTJ+1 (NUMBER OF SECOND DIMENSION FINE MESH BOUNDARIES)

-------------------------------------------------------
CR CELL BACK-EDGE ANGULAR FLUX

CL ((FBEDGE(I,J),I=NINTI),J=1,NINTJ)

Cw NINTI*NINTJ*MULT=NUMBER OF WORDS

CD FBEDGE(I,J) CELL BACK-EDGE ANGULAR FLUX

-------------------------------------------------------
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.

RMFLUX

DATE 04/01/85

ORDER OF GROUPS IS ACCORDING TO DECREASING ENERGY

FILE STRUCTURE

FILE IDENTIFICATION

HNAME, (HUSE(I), I=1,2), IVERS

1+3*MULT=NUMBER OF WORDS

HNAME

HOLLERITH FILE NAME - RMFLUX - (A6)

HNAME

HOLLERITH FILE NAME - (A6)

HUSE(I)

HOLLERITH USER IDENTIFICATION (A6)

IVERS

FILE VERSION NUMBER

MULT

DOUBLE PRECISION PARAMETER

1- A6 WORD IS SINGLE WORD

2- A6 WORD IS DOUBLE PRECISION WORD

SPECIFICATIONS (1D RECORD)

NDIM, NGROUP, NINTI, NINTJ, NINTK, NORD, EFFK, POWER, OITNO

9=NUMBER OF WORDS

NDIM

NUMBER OF DIMENSIONS

NGROUP

NUMBER OF GROUPS

NINTI

NUMBER OF FIRST DIMENSION INTERVALS

NINTJ

NUMBER OF SECOND DIMENSION INTERVALS

NINTK

NUMBER OF THIRD DIMENSION INTERVALS
CODE DEPENDENT INTERFACE FILES

RMFLUX

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----------------------------------------------------------------------

CR

CL REGULAR MOMENTS FLUXES ON MULTIDIMENSIONAL INTERVALS
   (2D RECORD)

CL ((FLUX(M,I),M=1,NORD),I=1,NINTI) ----NOTE STRUCTURE BELOW----

CW NORD*NINTI=NUMBER OF WORDS

DO 1 K=1,NINTK
   DO 1 J=1,NINTJ
      1 READ (N) *LIST AS ABOVE*

CD FLUX(M,I) REGULAR FLUX MOMENTS ON FIRST DIMENSION

CEOFRMLUX75
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.

CODE DEPENDENT INTERFACE FILES

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.

FILE IDENTIFICATION

HNAME, (HUSE(I), I=1, 2), IVERS

1+3*MULT=NUMBER OF WORDS

HNAME HOLLERITH FILE NAME - RZMFLX - (A6)
HNAME HOLLERITH FILE NAME - - (A6)
HUSE(I) HOLLERITH USER IDENTIFICATION (A6)
IVER FILE VERSION NUMBER
MULT DOUBLE PRECISION PARAMETER
  1- A6 WORD IS SINGLE WORD
  2- A6 WORD IS DOUBLE PRECISION WORD

SPECIFICATIONS (1D RECORD)

NDIM, NGROUP, NZONE, DUM, DUM, NORD, EFFK, POWER, OITNO

9=NUMBER OF WORDS

NDIM NUMBER OF DIMENSIONS
NGROUP NUMBER OF GROUPS
NZONE NUMBER OF GEOMETRIC ZONES
DUM DUMMY, NOT USED
DUM DUMMY, NOT USED
CODE DEPENDENT INTERFACE FILES

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  
CD FLUX(M,I) REGULAR FLUX MOMENT AVERAGES FOR EACH  --
CD ZONE.  --
C  
C-----------------------------------------------------------------------
CEOF  --

FILE DESCRIPTIONS  Version 3.0  15-77
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.
CODE DEPENDENT INTERFACE FILES

SNXEDT

CL   NGROUP, NISO, NORD, NED, IDPF, LNG, MAXUP, MAXDN, NPRIN, I2LP1

CW   10=NUMBER OF WORDS

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 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

-----------------------------------------------------------------------

FILE DATA

CL   (HISO(I), I=1, NISO), (HED(J), J=1, NEDT), (VEL(N), N=1, NGROUP),
CL   1 (EMAX(N), N=1, NGROUP), EMIN

CW   (NISO+NEDT+2*NGROUP+1)*MULT=NUMBER OF WORDS

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

CN   THE FOUR BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:

CN   HED(1) = 3HCHI
CN   HED(2) = 6HMUSIGF
CN   HED(3) = 5HTOTAL
CN   HED(4) = 3HABS

CN   ALSO PRESENT WHEN NPRIN=5 IS:
CN   HED(5) = 5HTRANS

-----------------------------------------------------------------------

PRINCIPAL CROSS SECTIONS FOR GROUP N

CL   ((C(I,J), I=1, NISO), J=1, NEDT)

CW   NISO*NEDT*MULT=NUMBER OF WORDS

CD   C(I,J) PRINCIPAL CROSS SECTIONS

CN   BASIC PRINCIPAL CROSS SECTIONS ALWAYS PRESENT ARE:

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

CN   ALSO PRESENT WHEN NPRIN=5 IS:

FILE DESCRIPTIONS Version 3.0 15-79
J=5 TRANSPORT CROSS SECTION

SCATTERING CONTROL BLOCK FOR GROUP N

PRESENT IF NORD.GT.0

((NGPB(L,J),L=1,NORD),J=1,NISO)

((IFSG(L,J),L=1,NORD),J=1,NISO)

2*NORD*NISO=NUMBER OF WORDS

NGPB(L,J) NUMBER OF SOURCE GROUPS THAT CAN SCATTER INTO GROUP N-

IFSG(L,J) GROUP NUMBER OF THE FIRST SOURCE GROUP

L LEGENDRE ORDER NUMBER

J ISOTOPE NUMBER

SCATTERING SUB-BLOCK FOR GROUP N

PRESENT IF NORD.GT.0

(SCAT(I),I=1,NTAB)

NTAB*MULT=NUMBER OF WORDS

SCAT(I) SCATTERING CROSS SECTION

NTAB TABLE LENGTH OF THE CROSS SECTIONS FOR SCATTERING

INTO GROUP N. THIS IS FOR ALL ISOTOPES AND ALL

LEGENDRE ORDERS, THUS IT IS THE SUM OF NGPB(L,J)

OVER L FROM 1 TO NORD AND OVER J FROM 1 TO NISO.

THE SCATTERING CROSS SECTIONS ARE PACKED IN BANDS,

ONE FOR EACH LEGENDRE ORDER AND ISOTOPE. EACH BAND

CONTAINS THE NGPB GROUPS WHICH SCATTER INTO GROUP

N. THE FIRST SOURCE GROUP NUMBER IS IFSG AND

THE LAST IS IFSG-NGPB+1. THE NORD BANDS FOR THE

FIRST ISOTOPE APPEAR FIRST (P0, P1, ...) FOLLOWED

BY THE NORD BANDS FOR THE SECOND, ETC.

HIGHER LEGENDRE ORDER SCATTERING CROSS SECTIONS

INCLUDE A 2*L+1 FACTOR WHERE L IS THE LEGENDRE

ORDER.

FILE DESCRIPTIONS  Version 3.0
The SOLINP code-dependent interface file contains information specific to the Solver Module, mainly the information from Block-V of the card-image input.

```plaintext
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 FILE STRUCTURE
C
RECORD TYPE PRESENT IF
-set-
FILE IDENTIFICATION ALWAYS
-set-
TITLE CARD COUNT ALWAYS
-set-
* TITLE CARD NHEAD.GT.0
-set-
*******
DIMENSION ALWAYS
-set-
RAW CONTROLS AND DIMENSIONS ALWAYS
-set-
RAW FLOATING INPUT DATA ALWAYS
-set-
DEFAULTED CONTROLS AND DIMENSIONS ALWAYS
-set-
DEFAULTED FLOATING INPUT DATA ALWAYS
-set-
BNDRY TRANSFER FIRST SOURCE GROUP IBL=4 OR IBR=4
-set-
BNDRY TRANSFER VECTOR LENGTHS IBL=4 OR IBR=4
-set-
*******(REPEAT FOR ALL GROUPS)
* LEFT BDRY TRANSFER VECTOR IBL=4
-set-
* RIGHT BDRY TRANSFER VECTOR IBR=4
-set-
*******
FINE MESH DENSITY FACTORS IDENX=2
-set-
FINE MESH DENSITY VECTOR IN X IDENX=1
-set-
FINE MESH DENSITY VECTOR IN Y IDENY=1
-set-
FINE MESH DENSITY VECTOR IN Z IDENZ=1
-set-
MONTE CARLO OPTION QUANTITIES MCOPT>0
-set-
ADAPTIVE WEIGHTED DIAMOND PARAMS ALWAYS
-set-
RADIUS MODIFIERS IN X IEVT=4.AND.IXM=1
-set-
RADIUS MODIFIERS IN Y IEVT=4.AND.IYM=1
-set-
RADIUS MODIFIERS IN Z IEVT=4.AND.IZM=1
-set-
SN ORDER BY GROUP IGRPSN=1
-set-
LEFT ALBEDOES IBEDEL.NE.0
-set-
RIGHT ALBEDOES IBEDEO.NE.0
-set-
BOTTOM ALBEDOES IBEDEL.NE.0
-set-
TOP ALBEDOES IBEDEO.NE.0
-set-
FRONT ALBEDOES IBEDEL.NE.0
-set-
BACK ALBEDOES IBEDEO.NE.0
-set-
SINGLE CHI ARRAY(FISSION SPECTRA) INCHI=1
-set-
*******(REPEAT FOR ALL ZONES)
* CHI ARRAY(FISSION SPECTRA) INCHI=2
-set-
*******
QUADRATURE WEIGHTS IQQUAD=3
-set-
QUADRATURE COSINES IQQUAD=3
-set-
*******(REPEAT FOR ALL MOMENTS)
* SOURCE SPECTRUM IQOPT=1
-set-
* SOURCE SPATIAL VECTOR(S) IQOPT=2
-set-
**********(REPEAT FOR ALL GROUPS)
-set-
```

FILE DESCRIPTIONS Version 3.0 15-81
CODE DEPENDENT INTERFACE FILES

SOLINP

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 **********

C-----------------------------------------------------------------------
C _______________________________________________________________________
CR FILE IDENTIFICATION
C
CL HNAME, (HUSE(I), I=1,2), IVERS
C
CW 1+3*WLT=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-----------------------------------------------------------------------

15-82
FILE DESCRIPTIONS Version 3.0
CODE DEPENDENT INTERFACE FILES

CR

SPatial DIMENSION

CW 1=NUMBER OF WORDS

CD IDIMEN=1 FOR ONEDANT

CR

RAW CONTROLS AND DIMENSIONS

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 IQQAD 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 11 IDENX 0/1/2 - NONE/FINE MESH DENSITY FACTORS BY XMESH/
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 IITM OUTER ITERATION LIMIT
CD 19 IITL EARLY INNER ITERATION LIMIT
CD 20 IITM NEAR CONVERGENCE INNER ITERATION LIMIT
CD 21 ITLIM TIME LIMIT IN SECONDS
CD 22 IL 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 FISSLR 0/1 - NO/YES FISSON RATE PRINT
CD 26 SOURCP 0/1/2/3/4/5 AO/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 31 IGPPSN GROUP DEP SN ORDERS (GRPSN) READ IN
CD 32 RAFLUX 0/1 - NO/YES WRITE ANGULAR FLUX FILE RAFLUX
CD 33 IBEDO ALBEDO OPTION
CD 34 IBALP 0/1 - NO/YES PRINT BALANCES BY ZONE

FILE DESCRIPTIONS Version 3.0
15-83
CODE DEPENDENT INTERFACE FILES

CD 35 DUM3 NOT USED
CD 36 IBB 0/1/2/3/4 BOTTOM BDRY CONDITION
CD 37 IFT 0/1/2/3/4 TOP BDRY CONDITION
CD 38 ITLD TIME LIMIT 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 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 51 IBK 0/1/2/3/4 BACK BDRY CONDITION
CD 52 MCNHIS NUMBER OF HISTORIES FOR MC OPTION
CD 53 MCNHSST 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 MCPRTN PRINT LEVEL FOR MC OPTION
CD 59 MCSEED REST INITIAL RANDOM SEED
CD 60 IMTSRC USED IN MC OPTION
CD 61 IMBSRA USED IN MC OPTION
CD 62 IMBSBA USED IN MC OPTION
CD 63 IMBSTA USED IN MC OPTION
CD 64 IMBIO
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 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 IBEDOF 0/1/2 NONE/FRONT ALBEDO/ALBEDO SPATIAL DISTRIBUTION
CD 71 IBEDOK 0/1/2 NONE/BACK ALBEDO/ALBEDO SPATIAL DISTRIBUTION
CD 72 NOSIGF 0/1 NONE/SET NUSGF ZERO FOR SOURCE PROBLEMS
CD 73 IFLANT 0/1 NONE/PLANET VARIABLE PRESENT
CD 74 JSRTE 0/1 = NO/YES WRITE RIGHT BOUNDARY FLUX AT i
CD 75 JSBOTT 0/1 = NO/YES WRITE BOTTOM BOUNDARY FLUX AT i
CD 76 JSTOP 0/1 = NO/YES WRITE TOP BOUNDARY FLUX AT i
CD 77 JSLEFT 0/1 = NO/YES WRITE LEFT BOUNDARY FLUX AT i
CD 78 JSLEFT 0/1 = 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 FCNRY 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

CR RAW FLOATING DATA
C
CL EV, NORM, EPSO, EPSI, BHGT, BWTH, EVM, PV, XLAX, XLAH,
CL 1 XLAX, POD, EPSR, EPSX, EPI

FILE DESCRIPTIONS Version 3.0
CODE DEPENDENT INTERFACE FILES

CL 2 (DUM(I),I=1,10),
CL 3 EFAC, TO, TS, EOM, SIGTH, TRCOR, PLANET, FCSRC, XMCSB, XMCBLT,
CL 4 (DUM(I),I=1,55),
CL 5 (EXTRAS(I),I=1,110)

CW 200*MULT=NUMBER OF WORDS

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 XALW LAMBDA LOWER LIMIT FOR SEARCHES
CD 10 XAHU LAMBDA UPPER LIMIT FOR SEARCHES
CD 11 XAX 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 EPSR PARAMETER OSCILLATION DAMPER
CD 16 DUM VECTOR NOT USED
CD 17 EFAC NOT USED (RESERVED FOR TIME DEPENDENCE)
CD 18 TO NOT USED (RESERVED FOR TIME DEPENDENCE)
CD 19 TS NOT USED (RESERVED FOR TIME DEPENDENCE)
CD 20 EOM NOT USED (RESERVED FOR TIME DEPENDENCE)
CD 21 SIGTH NOT USED (RESERVED FOR TIME DEPENDENCE)
CD 22 TRCOR TRANSPORT CORRECTION INDICATOR
CD 23 PLANET PLANET INDICATOR
CD 24 FCSRC USE FIRST COLLISION SOURCE OPTION
CD 25 XMCSB BIASING PARAMETER IN MC OPTION
CD 26 XMCLNT BOUNDARY LAYER THICKNESS IN MC OPTION
CD 27 FCWCO WEIGHT CUTOFF FOR FIRST COLLISION RAYS
CD 28 EXTRAS VECTOR USED BY INDIVIDUAL SOLVERS

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

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

CR BOUNDARY TRANSFER FIRST SOURCE GROUP
CC PRESENT IF IBL.EQ.4 .OR. IBR.EQ.4
CODE DEPENDENT INTERFACE FILES

CL (IFSGL(N), N=1, NGROUP), (IFSGR(N), N=1, NGROUP)

CW 2*NGROUP=NUMBER OF WORDS

CD IFSSL FIRST SOURCE GROUP FOR LEFT BOUNDARY
CD IFSSR FIRST SOURCE GROUP FOR RIGHT BOUNDARY
CD NGROUP NUMBER OF ENERGY GROUPS

CR BOUNDARY TRANSFER VECTOR LENGTHS
CC PRESENT IF IBL.EQ.4 OR IBR.EQ.4
CL (LENL(N), N=1, NGROUP), (LENR(N), N=1, NGROUP)
CW 2*NGROUP=NUMBER OF WORDS
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

CR LEFT BOUNDARY TRANSFER VECTOR
CC PRESENT IF IBL.EQ.4
CL (TRL(I), I=1, NW)
CW NW=NUMBER OF WORDS
CD TRL(I) I-TH TRANSFER INTO GROUP N FOR LEFT BOUNDARY
CD THE FIRST VALUE IS THE TRANSFER FROM GROUP IFSSL(N)
CD INTO GROUP N, NEXT IS FROM GROUP IFSSL(N)-1 TO N, ETC.
CD NW LENV(N)
CD N NUMBER OF THE RECEIVING GROUP

CR RIGHT BOUNDARY TRANSFER VECTOR
CC PRESENT IF IBR.EQ.4
CL (TRR(I), I=1, NW)
CW NW=NUMBER OF WORDS
CD TRR(I) I-TH TRANSFER INTO GROUP N FOR RIGHT BOUNDARY
CD THE FIRST VALUE IS THE TRANSFER FROM GROUP IFSSR(N)
CD INTO GROUP N, NEXT IS FROM GROUP IFSSR(N)-1 TO N, ETC.
CD NW LENV(N)
CD N NUMBER OF THE RECEIVING GROUP

CR FINE MESH DENSITY FACTORS
CC PRESENT IF IDENX.EQ.2
CODE DEPENDENT INTERFACE FILES

(CL (DEN(I),I=1,IT)----NOTE STRUCTURE BELOW----

(CW IT*MULT=NUMBER OF WORDS

CS DO 1 K=1,KT
CS DO 1 J=1,JT
CS 1 READ (N) *LIST AS ABOVE*

(CD IT NUMBER OF FINE MESH INTERVALS

------------------------------------------------------------------------

CR FINE MESH DENSITY VECTOR IN X

CC PRESENT IF IDENX.EQ.1

CL (DEN(I),I=1,IT)

(CW IT*MULT=NUMBER OF WORDS

CD IT NUMBER OF FINE MESH INTERVALS IN X DIRECTION

------------------------------------------------------------------------

CR FINE MESH DENSITY VECTOR IN Y

CC PRESENT IF IDENY.EQ.1

CL (DEN(J),J=1,JT)

(CW JT*MULT=NUMBER OF WORDS

CD JT NUMBER OF FINE MESH INTERVALS IN Y DIRECTION

------------------------------------------------------------------------

CR FINE MESH DENSITY VECTOR IN Z

CC PRESENT IF IDENZ.EQ.1

CL (DEN(K),K=1,KT)

(CW KT*MULT=NUMBER OF WORDS

CD KT NUMBER OF FINE MESH INTERVALS IN Z DIRECTION

------------------------------------------------------------------------

CR MONTE CARLO OPTION PARAMETERS

CC PRESENT IF MCOPT.EQ.1

CL (JBND(I),I=1,2)

(CW 2*MULT=NUMBER OF WORDS

CL (IBND(I),I=1,2)

(CW 2*MULT=NUMBER OF WORDS

CL (MCREG(G),G=1,IGM)
CODE DEPENDENT INTERFACE FILES

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

---------------------------------------------------------------------

CR     AWDD PARAMETERS
CC     PRESENT ALWAYS
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

---------------------------------------------------------------------

CR     RADIAL MODIFIER IN X
CC     PRESENT IF IEVT.EQ.4 .AND. IXM.EQ.1
CL     (RM(I),I=1,IM)
C CW     IM*MULT=NUMBER OF WORDS
C CD     IM         NUMBER OF COARSE MESH INTERVALS

---------------------------------------------------------------------

CR     RADIAL MODIFIER IN Y
CC     PRESENT IF IEVT.EQ.4 .AND. IYM.EQ.1
CODE DEPENDENT INTERFACE FILES

CODE DEPENDENT INTERFACE FILES

FILE DESCRIPTIONS Version 3.0 15-89
CODE DEPENDENT INTERFACE FILES

FILE DESCRIPTIONS  Version 3.0

15-90

NGROUP*MULT=NUMBER OF WORDS

PRESENT IF IBEDO.EQ.2

(RBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(BBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(TBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(TBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(BBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(RBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(RBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(BBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(RBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(BBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(TBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(TBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(RBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(RBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.1

(BBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*

------------------------------------------

PRESENT IF IBEDO.EQ.2

(BBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 K=1,KT
DO 1 J=1,JT
1 READ(N) *LIST AS ABOVE*
CODE DEPENDENT INTERFACE FILES

PRESENT IF IBEDOF.EQ.1

(FBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

PRESENT IF IBEDOF.EQ.2

(FBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 J=1,JT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

BACK ALBEDOES

PRESENT IF IBEDOK.EQ.1

(BKBEDO(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

PRESENT IF IBEDOK.EQ.2

(BKBEDO(N),N=1,NGROUP) ----NOTE STRUCTURE BELOW----

NGROUP*MULT=NUMBER OF WORDS

DO 1 J=1,JT
DO 1 I=1,IT
1 READ(N) *LIST AS ABOVE*

CHI - FISSION SPECTRA

PRESENT IF INCHI.NE.0

(CHI(N),N=1,NGROUP)

NGROUP*MULT=NUMBER OF WORDS

QUADRATURE WEIGHTS

PRESENT IF IQUAD.EQ.3

(WGT(M),M=1,MM)

MM*MULT=NUMBER OF WORDS

MM NUMBER OF ANGLES IN QUADRATURE SET
CODE DEPENDENT INTERFACE FILES

C CODE DEPENDENT INTERFACE FILES
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 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 CR SOURCE SPATIAL VECTOR IN FIRST DIMENSION
C CC PRESENT IF IQOPT.EQ.2 .OR. IQOPT.EQ.4
C CL (SOURCEX(I),I=1,IT)
C CW IT*MULT=NUMBER OF WORDS
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 (SOURCEY(J),J=1,JT)
C CW JT*MULT=NUMBER OF WORDS
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 (SOURCEZ(K),K=1,KT)
C CW KT*MULT=NUMBER OF WORDS
C ______________________________________________________________________
C CR SOURCE POINTWISE FULL DISTRIBUTION
C CC PRESENT IF IQOPT.EQ.3 .OR. IQOPT.EQ.6
C CL (SOURCEX(I),I=1,IT)--NOTE STRUCTURE BELOW-----
C CW IT*MULT=NUMBER OF WORDS

FILE DESCRIPTIONS Version 3.0
CODE DEPENDENT INTERFACE FILES

DO 1 K=1,KT
CS DO 1 J=1,JT
CS 1 READ(N) *LIST AS ABOVE*
C
C

-----------------------------------------------------------------------

CR
C
CC BOUNDARY ISOTROPIC SOURCE (LEFT SHOWN)
CC
CL PRESENT IF IBL.EQ.-1 .AND. IDIMEN.EQ.1 .AND. GROUP.EQ.1
CL (SILEFT(N),N=1,NGROUP)
C
CW NGROUP*MULT=NUMBER OF WORDS
CD SILEFT(N) LEFT BOUNDARY SOURCE FOR GROUP N
C
C
CC PRESENT IF IBL.EQ.-1 .AND. IDIMEN.GT.1
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 INTERVAL (J,K)
CD JT NUMBER OF FINE MESH INTERVALS IN THE SECOND DIMENSION. EQUALS 1 FOR ONEDANT.
CD KT NUMBER OF FINE MESH INTERVALS IN THE THIRD DIMENSION. EQUALS 1 FOR ONEDANT AND TWODANT.
C
C

-----------------------------------------------------------------------

CR
C
CC BOUNDARY ANISOTROPIC SOURCE (LEFT SHOWN)
CC
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 THE GROUP AT INTERVAL (J,K)
CD MMHALF NUMBER OF INCOMING ANGLES IN THE SOURCE
C

-----------------------------------------------------------------------

CR
C
CC BOUNDARY SOURCE SPECTRUM (LEFT SHOWN)
CC
CL (SOURCE(N),N=1,NGROUP)
C
C

FILE DESCRIPTIONS  Version 3.0  15-93
CODE DEPENDENT INTERFACE FILES

CW NGROUP*MULT=NUMBER OF WORDS
C
C----------------------------------------------------------------------

CR BOUNDARY SOURCE SPATIAL DISTRIBUTION VECTOR 1 (LEFT SHOWN)
C
CC PRESENT IF IQL.EQ.2 .AND. GROUP.EQ.1

CL (SOURCE(J),J=1,JT)

CW JT*MULT=NUMBER OF WORDS
C
C----------------------------------------------------------------------

CR BOUNDARY SOURCE SPATIAL DISTRIBUTION VECTOR 2 (LEFT SHOWN)
C
CC PRESENT IF IQL.EQ.2 .AND. GROUP.EQ.1

CL (SOURCEZ(K),K=1,KT)

CW KT*MULT=NUMBER OF WORDS
C
C----------------------------------------------------------------------

CR BOUNDARY SOURCE SPACE ARRAY (LEFT SHOWN)
C
CC PRESENT IF IQL.EQ.3 .AND. GROUP.EQ.1

CL (SOURCE(J),J=1,JT) ----NOTE STRUCTURE BELOW----

CW JT*MULT=NUMBER OF WORDS
C
CS DO 1 K=1,KT
CS 1 READ(N) *LIST AS ABOVE*
C
C----------------------------------------------------------------------

CR BOUNDARY SOURCE SPACE-ANGLE ARRAY (LEFT SHOWN)
C
CC PRESENT IF IQL.EQ.4 .AND. GROUP.EQ.1

CL (SOURCE(J),J=1,JT) ----NOTE STRUCTURE BELOW----

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----------------------------------------------------------------------

CR BOUNDARY SOURCE ANGLE-SPACE ARRAY (LEFT SHOWN)
C
CC PRESENT IF IQL.EQ.5 .AND. GROUP.EQ.1

CL (SALEFT(M),M=1,MMHALF) ----NOTE STRUCTURE BELOW----

CW MMHALF*MULT=NUMBER OF WORDS
C
CS DO 1 K=1,KT
CODE DEPENDENT INTERFACE FILES

CS   DO 1 J=1, JT
CS 1 READ(N) *LIST AS ABOVE*
C
C---------------------------------------------------------------
C
CEOFSOLINP
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.

File Structure:

- RECORD TYPE: PRESENT IF
- INT SPECIFICATIONS: ALWAYS
- FP SPECIFICATIONS: ALWAYS
- NUMBER OF TRACKS: I(2.GT.0)
- INTEGRAL FLUXES: I(2.GT.0)

(REPEAT FOR NSGRP GROUPS)

- UNCOLLIDED FLUX MOMENTS: ALWAYS

NSGRP IS THE NUMBER OF GROUPS WITH AN INHOMOGENEOUS SOURCE

INT SPECIFICATIONS (1D RECORD)

- NHSTOT, NXYSCORE, IPREVTR, NWDS

4 = NUMBER OF WORDS

- NHSTOT: NUMBER OF RAYS TRACED
- NXYSCORE: NUMBER OF RAYS TERMINATED
- IPREVTR: NUMBER OF PREVIOUS TRIALS
- NWDS: TOTAL NUMBER OF WORDS IN FILE
CODE DEPENDENT INTERFACE FILES

FP SPECIFICATIONS (1D RECORD)

WTMIN, WTMAX, WTTOT, PTIME

4 = NUMBER OF WORDS

WTMIN
  MINIMUM RAY WEIGHT
WTMAX
  MAXIMUM RAY WEIGHT
WTTOT
  TOTAL RAY WEIGHT
PTIME
  CPU TIME

NUMBER OF TRACKS (2D RECORD)

PRESENT IF I2.GT.0

((NTENTR(I,J), I=1, IT), J=1, JT)

IT*JT = NUMBER OF WORDS

NTENTR(I,J)
  NUMBER OF RAYS ENTERING EACH FINE MESH

INTEGRAL FLUXES (2D RECORD)

PRESENT IF I2.GT.0

((FLRT(I,J), I=1, IT), J=1, JT), ((FLRT2(I,J), I=1, IT), J=1, JT)

IT*JT*2 = NUMBER OF WORDS

FLRT(I,J)
  INTEGRAL FLUXES FOR VARIANCE CALC
FLRT2(I,J)
  INTEGRAL FLUXES**2 FOR VARIANCE CALC

LEAKAGES (1D RECORD)

((FCRL(I), I=1, IGM), (FCLL(I), I=1, IGM), (FCTL(I), I=1, IGM),
(FCBL(I), I=1, IGM)

IGM*4 = NUMBER OF WORDS

FCRL(I)
  FCSRC RIGHT LEAKAGE (UNNORMALIZED)
FCLL(I)
  FCSRC LEFT LEAKAGE (UNNORMALIZED)
FCTL(I)
  FCSRC TOP LEAKAGE (UNNORMALIZED)
FCBL(I)
  FCSRC BOTTOM LEAKAGE (UNNORMALIZED)

UNCOLLIDED FLUX MOMENTS (3D RECORD)

((FLUC(I,J,N), I=1, IT), J=1, JT), N=1, NM

FILE DESCRIPTIONS
Version 3.0
15-97
CODE DEPENDENT INTERFACE FILES

UCFLUX

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


CODE ABSTRACTS

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

16
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 mainframe 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 $S_4$ 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 quadra-
ture, and a $14\times14\times30$ 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.
Deterministic Transport Team
Transport Methods Group, XTM
Los Alamos National Laboratory


**BIBLIOGRAPHY Version 3.0**


A

Acceleration
Chebychev .................................................. 5-11
Diffusion synthetic(DSA) ............................................... 12-14
Upscatter(Grey) .......................................................... 2-49, 3-51, 7-21
ACCELERATION DISABLED message .............................. 7-22
Adaptive weighted diamond differencing .......................... 3-53, 6-53

Adjoint
Calculation of .......................................................... 7-37
Edit of .............................................................. 2-65, 3-72, 4-75, 5-57, 6-69, 8-15
Group order in ...................................................... 2-28, 3-30, 4-34, 5-28, 6-30, 7-38

Albedo
Input ................................................................. 2-57
Use of ............................................................... 2-48, 7-24

Alpha(time absorption) search ........................................... 1-11, 7-34

Angular flux
File output ........................................................... 2-50, 3-52, 4-60, 6-52
Input of boundary ...................................................... 2-56, 3-59, 4-67, 6-58
Print output ........................................................... 2-50, 3-52, 4-60, 6-52
Short output file ....................................................... 3-52

Anisotropy
Cross section scattering source ........................................ 12-22

Approximation
Discrete ordinates ..................................................... 12-12
Multigroup ............................................................. 12-11
Transverse leakage ...................................................... 7-33

Array
Definition of .......................................................... 2-19, 3-19, 4-25, 5-19, 6-19, 9-9
Notation for order .................................................... 2-22, 3-22, 4-28, 5-22, 6-22
Notation for size ....................................................... 2-22, 3-22, 4-28, 5-22, 6-22

ASSIGN
Example of use ........................................................ 11-11
Input template ........................................................ 2-43, 3-45, 4-54, 5-42, 6-45

B

Block
Definition of .......................................................... 2-20, 3-20, 4-26, 5-20, 6-20, 9-10
Order in input ........................................................... 2-17, 3-17, 4-23, 5-17, 6-17

Block-I input
ONEDANT .......................................................... 2-31
THREEDANT ........................................................ 6-33
TWODANT ............................................................ 3-33
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWODANT/GQ</td>
<td>4-37</td>
</tr>
<tr>
<td>TWOHEX</td>
<td>5-31</td>
</tr>
<tr>
<td>Block-II input</td>
<td></td>
</tr>
<tr>
<td>ONEDANT</td>
<td>2-34</td>
</tr>
<tr>
<td>THREEEDANT</td>
<td>6-36</td>
</tr>
<tr>
<td>TWODANT</td>
<td>3-36</td>
</tr>
<tr>
<td>TWODANT/GQ</td>
<td>4-39</td>
</tr>
<tr>
<td>TWOHEX</td>
<td>5-33</td>
</tr>
<tr>
<td>Block-III input</td>
<td></td>
</tr>
<tr>
<td>ONEDANT</td>
<td>2-35</td>
</tr>
<tr>
<td>THREEEDANT</td>
<td>6-37</td>
</tr>
<tr>
<td>TWODANT</td>
<td>3-37</td>
</tr>
<tr>
<td>TWODANT/GQ</td>
<td>4-46</td>
</tr>
<tr>
<td>TWOHEX</td>
<td>5-34</td>
</tr>
<tr>
<td>Block-IV input</td>
<td></td>
</tr>
<tr>
<td>ONEDANT</td>
<td>2-41</td>
</tr>
<tr>
<td>THREEEDANT</td>
<td>6-43</td>
</tr>
<tr>
<td>TWODANT</td>
<td>3-43</td>
</tr>
<tr>
<td>TWODANT/GQ</td>
<td>4-52</td>
</tr>
<tr>
<td>TWOHEX</td>
<td>5-40</td>
</tr>
<tr>
<td>Blocks</td>
<td></td>
</tr>
<tr>
<td>Input order</td>
<td>2-17, 3-17, 4-23, 5-17, 6-17</td>
</tr>
<tr>
<td>Block-V input</td>
<td></td>
</tr>
<tr>
<td>ONEDANT</td>
<td>2-48</td>
</tr>
<tr>
<td>THREEEDANT</td>
<td>6-50</td>
</tr>
<tr>
<td>TWODANT</td>
<td>3-50</td>
</tr>
<tr>
<td>TWODANT/GQ</td>
<td>4-59</td>
</tr>
<tr>
<td>TWOHEX</td>
<td>5-45</td>
</tr>
<tr>
<td>Block-VI input</td>
<td></td>
</tr>
<tr>
<td>ONEDANT</td>
<td>2-58</td>
</tr>
<tr>
<td>THREEEDANT</td>
<td>6-61</td>
</tr>
<tr>
<td>TWODANT</td>
<td>3-65</td>
</tr>
<tr>
<td>TWODANT/GQ</td>
<td>4-68</td>
</tr>
<tr>
<td>TWOHEX</td>
<td>5-50</td>
</tr>
<tr>
<td>Boltzmann transport equation</td>
<td>12-9</td>
</tr>
<tr>
<td>Boundary condition</td>
<td></td>
</tr>
<tr>
<td>Definition of, on a segment</td>
<td>4-17</td>
</tr>
<tr>
<td>Discussion of</td>
<td>7-23</td>
</tr>
<tr>
<td>Options</td>
<td>2-48, 3-50, 4-21, 4-59, 6-50</td>
</tr>
<tr>
<td>Boundary segment</td>
<td></td>
</tr>
<tr>
<td>Boundary condition on</td>
<td>4-44</td>
</tr>
<tr>
<td>Definition of</td>
<td>4-22</td>
</tr>
<tr>
<td>Example of</td>
<td>4-22</td>
</tr>
<tr>
<td>Boundary source</td>
<td>2-56, 3-59, 4-67, 6-58, 7-28</td>
</tr>
<tr>
<td>Buckling</td>
<td></td>
</tr>
<tr>
<td>Simulated leakage</td>
<td>2-51, 3-53, 4-61, 5-46, 7-33</td>
</tr>
</tbody>
</table>
### Character names
- In mixing arrays: 2-44, 3-46, 4-55, 5-43, 6-46

### Chebychev acceleration
- 5-11

### Chi
- Choice from MENDF: 2-36, 3-38, 4-47, 5-35, 6-38
- Isotope independent input: 2-36, 3-38, 4-47, 5-35, 6-38
- Zone dependent: 2-51, 3-53, 4-61, 5-46, 6-53, 7-25

### Coarse mesh
- Definition of: 1-10, 7-13

### Comments
- Embedded in input lines: 2-20, 3-20, 4-26, 5-20, 6-20, 9-10

### Component
- Definition of: 4-17
- Example of: 4-20
- Input for: 4-42

### Concentration search, see Searches, Concentration ...

### Convergence
- Behavior in search: 7-37
- Iterating to: 1-11, 7-15
- Tests: 7-19

### Convergence controls
- Special for criticality: 2-50, 3-51, 6-51

### Cross section library
- ASCII, writing from code: 10-19, 10-21
- Balancing of: 2-36, 3-38, 4-47, 5-35, 6-38
- Binary, writing from code: 10-12
- BXSLIB: 10-12
- Card image libraries: 10-9
- Converting to ASCII: 2-36, 3-37, 4-46, 5-34, 6-38
- Coupled neutron-gamma libraries: 10-15
- Details for inputting: 10-1
- Edit names on MENDF: 2-66, 3-74, 4-77, 5-58, 6-71
- Edit positions and names on: 2-60, 3-67, 4-70, 5-52, 6-64
- GRUPXS: 10-9
- ISOTXS: 10-9
- MACBCD: 10-13
- MACRCS: 10-13
- MENDF5: 10-13
- MENDF5G: 10-14
- SNXEDT: 10-13
- Specifying what, where: 2-35, 3-37, 4-46, 5-34, 6-37
- Text, format of: 2-39, 3-41, 4-50, 5-38, 6-41
- Text, order within: 2-40, 3-42, 4-51, 5-39, 6-42
- Text, position in input stream: 2-17, 3-17, 4-23, 5-17, 6-17
Transport correcting .................................................. 2-51, 3-53, 4-61, 6-53, 7-31
Writing ASCII library from code .............................. 2-36, 3-37, 4-46, 5-34, 6-38
XSLIB ............................................................................................................. 10-9
XSLIBB ........................................................................................................ 10-12
XSLIBE, XSLIFB ........................................................................................ 10-14

D

Data item
Character, definition of .....................................2-19, 3-19, 4-25, 5-19, 6-19, 9-10
Numeric, definition of ........................................2-19, 3-19, 4-25, 5-19, 6-19, 9-9
Data operators
Purpose of .................................................................2-20, 3-20, 4-26, 5-20, 6-20
Summary table of ...................................................... 2-21, 3-21, 4-27, 5-21, 6-21
Usage form ............................................................... 2-20, 3-20, 4-26, 5-20, 6-20
Delimiters
In free field input .......................................................... 9-13
Density factors
In edits ........................................................................2-59, 3-66, 4-69, 5-51, 6-63
In the flux calculation .................................................... 2-51, 3-53, 4-61, 5-46, 6-53
Diffusion synthetic acceleration(DSA)
General method .......................................................... 12-14
Dimension search, see Searches, Dimension ...
Discrete ordinates approximation .............................. 12-12
Discrete ordinates equation
In one dimension .......................................................... 12-27
Discretization of the spatial variable
In one dimension .......................................................... 12-32
In triangular geometry .............................................. 12-51
In two dimensions .......................................................... 12-37
In TWOHEX .......................................................... 12-51
Divergence operator in one dimension .................................................. 12-21
Documentation available
For ONEDANT .......................................................... 2-13
For THREEDANT ...................................................... 6-13
For TWODANT .......................................................... 3-13
For TWODANT/GQ ..................................................... 4-13
For TWOHEX .......................................................... 5-13

E

Edit Module
Function of .......................................................... 13-17
Edit names on MENDF ............................................. 2-66, 3-74, 4-77, 5-58, 6-71
Edit positions and names
Table of .................................................................................................................. 2-60, 3-67, 4-70, 5-52, 6-64

Edits
- Energy specifications ................................................................. 2-62, 3-69, 4-72, 5-54, 6-66, 8-10
- Reaction rates
  - From cross sections ........................................ 2-59, 3-66, 4-69, 5-51, 6-62, 8-11
  - From user defined response functions ........ 2-61, 3-68, 4-71, 5-53, 6-65, 8-13
- Spatial specifications .......................................................... 2-58, 3-65, 4-68, 5-50, 6-61, 8-9

Energy groups
- Broad groups in edits ........................................ 2-62, 3-69, 4-72, 5-54, 6-66, 8-10
- Order in adjoint run ........................................ 2-28, 3-30, 4-34, 5-28, 6-30, 7-38

Error messages .......................................................................................... 14-5

Execution of code
- Multiple runs ................................................................................. 13-25
- On UNIX systems ................................................................. 2-93, 3-123, 4-95, 5-75, 6-97
- Piecewise ....................................................................................... 13-19

Expansion of the inhomogeneous source ..................................................... 12-26
Expansion of the scattering source ............................................................. 12-22
Extraneous source, see Source, inhomogeneous

F

FIDO ................................................................................................................. 9-19

File description
- AAFLXM for THREEDANT ......................................................... 15-24
- AAFLXM for TWODANT .......................................................... 15-22
- ADJMAC ......................................................................................... 15-28
- AMFLUX .......................................................................................... 15-31
- ARBFLUX ......................................................................................... 15-17
- ASGMAT .......................................................................................... 15-33
- AZMFLX .......................................................................................... 15-36
- BXSLIB ............................................................................................. 15-38
- EDITIT ................................................................................................ 15-40
- EDTOGX ......................................................................................... 15-13
- EDTOUT ............................................................................................ 15-7
- FISSRC ............................................................................................... 15-47
- GEODST, extended ................................................................. 15-49
- GEOSING ......................................................................................... 15-61
- LNK3DNT ......................................................................................... 15-62
- MACRXS ......................................................................................... 15-64
- RAFLXM for THREEDANT ......................................................... 15-70
- RAFLXM for TWODANT .............................................................. 15-67
- RMFLUX .......................................................................................... 15-74
- RZMFLX ........................................................................................ 15-76
- SNXEDT ......................................................................................... 15-78
- SOLINP ......................................................................................... 15-81
- UCFLUX ......................................................................................... 15-96

Version 3.0
Files
Detailed descriptions of ................................................................. 15-1
Input that will be overwritten ....................................................... 15-99
Output, control of 2-50, 2-65, 3-52, 3-72, 4-60, 4-75, 5-46, 5-57, 6-52, 6-69, 8-15
Standard interface ........................................................................ 13-7
Suppressing writing .......................................................... 2-33, 3-35, 4-38, 5-32, 6-35, 13-20
Used in mixing ........................................................................... 11-17
Where read, written ..................................................................... 13-9
Fine mesh
Definition of ................................................................. 1-10, 7-13
Fine mesh mixing option .................................................. 3-48, 6-48
First collision source options
Analytic .................................................................................... 3-60
Ray tracing .................................................................................. 3-60
Fission fraction
Choice from MENDF .................................................. 2-36, 3-38, 4-47, 5-35, 6-38
Isotope independent input ........................................ 2-36, 3-38, 4-47, 5-35, 6-38
Zone dependent ................................................ 2-51, 3-53, 4-61, 5-46, 6-53, 7-25
Fixed field format ........................................................................ 9-19
Flow, control .............................................................................. 1-13
Flow, data .................................................................................... 1-13
Flux
Input guess from file ....................................................... 2-52, 3-55, 4-63, 5-47, 6-55
Moments file output ................................................... 2-50, 3-52, 4-60, 6-52
Normalization of ............................................................. 2-51, 3-53, 4-61, 5-46, 6-53, 7-30
Print control ............................................................................. 2-50, 3-52, 4-60, 5-46, 6-52
Free field input
Delimiters ..................................................................................... 9-13
Summary ................................................................................. 2-19, 3-19, 4-25, 5-19, 6-19
Syntax details .............................................................................. 9-13
Terminators ................................................................................ 9-13
User specified format ............................................................... 9-16
Geometry
Coarse mesh .............................................................................. 7-13
Concepts in TWODANT/GQ ...................................................... 4-17
Definition of TWODANT/GQ GEOMETRY array ......................... 4-17
Example of TWODANT/GQ GEOMETRY array ......................... 4-20
Fine mesh ................................................................................. 7-13
Input arrays .............................................................................. 2-34, 3-36, 4-39, 5-33, 6-36
Input for TWODANT/GQ GEOMETRY array ................................ 4-43
Symmetries treated in ONEDANT ........................................... 12-19
Grey acceleration ...................................................................... 2-49, 3-51, 7-21
Group collapse

INDEX-8 Version 3.0
Chapter on solution methods ................................................................. 12-1
Diffusion synthetic ................................................................................... 12-14
Monte Carlo/Discrete Ordinates ............................................................... 12-40

MINI-MANUAL
  Definition and purpose .............................................................. 2-22, 3-22, 4-28, 5-22, 6-22
  Graphic of ...................................................................................... 2-23, 3-23, 4-29, 5-23, 6-23

Mixing
  ASSIGN input array example .......................................................... 11-11
  Concepts in .................................................................................... 2-42, 3-44, 4-53, 5-41, 6-44
  Fine mesh input option .................................................................. 3-48, 6-48
  Materials ......................................................................................... 2-42, 3-44, 4-53, 5-41, 6-44
  MATLS input array example ............................................................ 11-9
  Premixes ......................................................................................... 2-42, 3-44, 4-53, 5-41, 6-44
  Terminology ..................................................................................... 11-7
  Tutorial ............................................................................................. 11-1
  Using atomic fractions .................................................................... 11-13
  Using weight fractions .................................................................... 11-13
  While assigning materials to zones .................................................. 11-11

Modules
  Suppressing execution of ............................................................... 2-33, 3-35, 4-38, 5-32, 6-35

Monitor
  Iteration, print of ............................................................................ 7-21

Monte Carlo/Discrete Ordinates
  Input ................................................................................................... 3-61
  Methods ............................................................................................. 12-40
  Sample problem ............................................................................. 3-97

Multigroup approximation .................................................................. 12-11

Multiple runs ..................................................................................... 13-25

NEG. SOURCE - ACCELERATION DISABLED message ......................... 7-22
NEG. SOURCE - TRANSPORT FLUXES BAD message ......................... 7-23

Normalization
  Of edit reaction rates ..................................................................... 2-64, 3-71, 4-74, 5-56, 6-68
  Of the flux ...................................................................................... 2-51, 3-53, 4-61, 5-46, 6-53, 7-30
  Of the source rate .......................................................................... 2-51, 3-53, 4-61, 5-46, 6-53, 7-30

Notation
  For expected order within array .................................................. 2-22, 3-22, 4-28, 5-22, 6-22
  For expected size of array ............................................................ 2-22, 3-22, 4-28, 5-22, 6-22

Numeric names
  In mixing arrays ................................................................. 2-44, 3-46, 4-55, 5-43, 6-46
O
Object
Definition of ................................................................................................ 4-17
Examples of .................................................................................................... 4-19
Input for .......................................................................................................... 4-41
Operator
Divergence operator in one dimension ......................................................... 12-21
Transport, in two dimensions ....................................................................... 12-36
Operators
In input, see also "Data operators" ............................................................. 2-20, 3-20, 4-26, 5-20, 6-20, 9-10
Summary table of .......................................................................................... 2-21, 3-21, 4-27, 5-21, 6-21
Optional marking on input arrays .............................................................. 2-27, 3-29, 4-33, 5-27, 6-29
Output
Angular flux file ............................................................................................ 2-50, 3-52, 4-60, 6-52
Angular flux print ........................................................................................... 2-50, 3-52, 4-60, 6-52
Controls .......................................................................................................... 2-50, 3-52, 4-60, 5-46, 6-52
Printed, control of contents ........................................................................... 2-50, 3-52, 5-46, 6-52
Overwriting input files .................................................................................... 15-99

P
Piecewise execution of code ........................................................................ 13-19
PREMIX
Input template ............................................................................................... 2-43, 3-45, 4-54, 5-42, 6-45
Purpose of ........................................................................................................ 2-43, 3-45, 4-54, 5-42, 6-45
Printed output
Control of contents ......................................................................................... 2-50, 3-52, 4-60, 5-46, 6-52

Q
Quadrature points
Number in one dimensional geometries .......................................................... 12-27
Number in two dimensions ............................................................................ 12-36
Ordering in one dimension ............................................................................ 12-28, 12-29, 12-30
Starting directions ........................................................................................ 12-32
Quadrature sets
Group dependent .......................................................................................... 3-54, 4-62, 6-54
ONEDANT, choices of ............................................................................... 2-52, 7-24
THREEEDANT, choices of .......................................................................... 6-54
TWODANT, choices of ............................................................................. 3-54
TWODANT/GQ, choices of ......................................................................... 4-62
TWOHEX, choices of .................................................................................. 5-47
R

Required marking on input arrays ................................... 2-27, 3-29, 4-33, 5-27, 6-29
Resolution, spatial ................................................................. 4-39
Response function
   Edits of ........................................................................... 2-61, 3-68, 4-71, 5-53, 6-65, 8-13
Run highlights ........................................................................... 14-15

S

Sample problem
  ONEDANT
     Description ........................................................................... 2-69, 2-86
     Input .................................................................................. 2-74, 2-88
     Output .................................................................................. 2-75, 2-88
  THREEDANT
     Description ........................................................................... 6-75
     Input .................................................................................. 6-77
     Output .................................................................................. 6-75
  TWODANT
     Description ........................................................................... 3-77, 3-97
     Input .................................................................................. 3-79, 3-102
     Output ................................................................................. 3-77, 3-97
  TWODANT/GQ
     Description ........................................................................... 4-81
     Input .................................................................................. 4-85
     Output .................................................................................. 4-82
  TWOHEX
     Description ........................................................................... 5-61
     Input .................................................................................. 5-64
     Output .................................................................................. 5-62

Searches
  Concentration
     Mixing arrays required .................................................. 2-44, 2-45, 3-46, 3-47, 4-55, 4-56, 6-46, 6-47
     Solver arrays required ..................................................... 2-54, 3-56, 4-64, 6-56
  Convergence behavior .......................................................... 7-37
  Dimension, arrays required .................................................. 2-53, 3-56, 4-64, 6-56
  General control of ................................................................. 2-53, 3-55, 4-63, 6-55, 7-33
  Time absorption(alpha) .......................................................... 1-11, 7-34
  Sn order
     Group dependent .................................................................. 3-54, 4-62, 6-54

Solution
  Chapter on solution methods ............................................... 12-1
  Two-angle plane ................................................................. 2-51

Solver Module
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function of</td>
<td>13-17</td>
</tr>
<tr>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>Anisotropy in</td>
<td>12-22</td>
</tr>
<tr>
<td>Boundary angular fluxes</td>
<td>2-56, 3-59, 4-67, 6-58, 7-28</td>
</tr>
<tr>
<td>First collision</td>
<td>3-60</td>
</tr>
<tr>
<td>Inhibit fission multiplication</td>
<td>2-48, 3-51, 6-51</td>
</tr>
<tr>
<td>Inhomogeneous, input of</td>
<td>7-26</td>
</tr>
<tr>
<td>Normalization of</td>
<td>2-51, 3-53, 4-61, 5-46, 6-53, 7-30</td>
</tr>
<tr>
<td>Volumetric, input of</td>
<td>2-55, 3-57, 4-65, 5-49, 6-57</td>
</tr>
<tr>
<td>Spatial discretization</td>
<td></td>
</tr>
<tr>
<td>Adaptive weighted diamond method (AWDD)</td>
<td>12-38</td>
</tr>
<tr>
<td>Diamond method</td>
<td>12-38</td>
</tr>
<tr>
<td>In one dimension</td>
<td>12-32</td>
</tr>
<tr>
<td>In two dimensions</td>
<td>12-37</td>
</tr>
<tr>
<td>Spatial resolution</td>
<td>4-39</td>
</tr>
<tr>
<td>Spherical harmonics</td>
<td></td>
</tr>
<tr>
<td>Expansion of the inhomogeneous source</td>
<td>12-26</td>
</tr>
<tr>
<td>Expansion of the scattering source</td>
<td>12-22</td>
</tr>
<tr>
<td>In one dimensional geometries</td>
<td>12-25</td>
</tr>
<tr>
<td>In two dimensions</td>
<td>12-35</td>
</tr>
<tr>
<td>Number in one dimensional geometries</td>
<td>12-24</td>
</tr>
<tr>
<td>Stacked runs</td>
<td>13-25</td>
</tr>
<tr>
<td>Storage</td>
<td></td>
</tr>
<tr>
<td>Memory requirements</td>
<td>2-32, 3-33, 4-38, 5-32, 6-34, 7-11</td>
</tr>
<tr>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Definition of</td>
<td>2-20, 3-20, 4-26, 5-20, 6-20, 9-10</td>
</tr>
<tr>
<td>Structure</td>
<td></td>
</tr>
<tr>
<td>Of code</td>
<td>1-13, 13-9</td>
</tr>
<tr>
<td>Submesh</td>
<td></td>
</tr>
<tr>
<td>Definition of</td>
<td>4-17</td>
</tr>
<tr>
<td>Examples of</td>
<td>4-18</td>
</tr>
<tr>
<td>Input for</td>
<td>4-40</td>
</tr>
<tr>
<td>Summing</td>
<td></td>
</tr>
<tr>
<td>Of reaction rates</td>
<td>2-63, 3-70, 4-73, 5-55, 6-67</td>
</tr>
<tr>
<td>Over energy in edits</td>
<td>2-62, 3-69, 4-72, 5-54, 6-66</td>
</tr>
<tr>
<td>Symbiosis</td>
<td></td>
</tr>
<tr>
<td>Monte Carlo and Discrete Ordinates</td>
<td>3-61</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>Terminating run</td>
<td>2-49, 3-51, 5-45, 6-51, 7-20</td>
</tr>
<tr>
<td>Terminators</td>
<td></td>
</tr>
<tr>
<td>In free field input</td>
<td>9-13</td>
</tr>
<tr>
<td>Theory, see Methods</td>
<td></td>
</tr>
<tr>
<td>Time absorption (alpha) search</td>
<td>1-11, 7-34</td>
</tr>
<tr>
<td>Time limit</td>
<td>2-49, 3-51, 5-45, 6-51</td>
</tr>
</tbody>
</table>
Input for ................................................................. 4-41
This report has been reproduced directly from the best available copy.

It is available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831. Prices are available from (615) 576-8401.

It is available to the public from the National Technical Information Service, US Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.